Estimating Generalized Gaussian Blur Kernels for Out-of-Focus Image Deblurring

Yu-Qi Liu, Xin Du, Hui-Liang Shen, *Member, IEEE*, and Shu-Jie Chen

***Abstract*— Out-of-focus blur is a common image degradation phenomenon that occurs in case of lens defocusing. The out-of-focus blur kernel is usually modeled as a Gaussian function or a uniform disk in previous work. In this paper, we propose that it can be more accurately depicted using the generalized Gaussian (GG) function. This is motivated by the theoretical analysis of the out-of-focus blur and the practical observation of real blur kernels. We show that as the out-of-focus blur kernels are of specific shapes, the GG function can be further simplified to a single-parameter model. We estimate the parameter of the GG blur kernel from image patches containing step edges, and obtain the clear image by non-blind image deblurring. Experimental results validate that the proposed GG blur kernel estimation algorithm outperforms the state-of-the-art ones deploying either parametric (disk and Gaussian) or nonparametric kernels, and consequently benefits the image deblurring process.**

***Index Terms*— Out-of-focus blur, image deblurring, generalized Gaussian function, parametric blur kernel, uniform disk kernel, Gaussian kernel.**

I. INTRODUCTION

IMAGE deblurring aims to recover the sharp image from a blurred observation. In general, the blurred images are caused by camera movement or lens defocusing. The ubiquitous image degradation usually leads to difficulties in object identification and scene interpretation. In this regard, image deblurring plays a prominent role in image processing and computer vision applications such as image reconstruction, biomedical imaging, face recognition, and space exploration [1]–[4].

In mathematical formulation, the observed blurred image $\mathbf{f}$ can be modeled as the convolution of the latent sharp image $\mathbf{g}$ with the point spread function (PSF) $\mathbf{h}$, resulting in

$$\mathbf{f} = \mathbf{g} * \mathbf{h} + \mathbf{n}, \tag{1}$$

where $\mathbf{n}$ is the image noise. The blind image deblurring problem is highly ill-posed as it needs to recover two unknowns, i.e., $\mathbf{g}$ and $\mathbf{h}$, from one observation $\mathbf{f}$. Recently, numerous blind image deblurring algorithms have been introduced to compute the latent sharp image and the blur kernel, by using either multiple images [5]–[7] or a single image with suitable image priors [8]–[10]. We note that, due to its ill-posed nature, it is difficult to find an effective and universal solution to the blind image deblurring problem. Fortunately, parametric blur kernels exist under some circumstances such as linear motion blur and out-of-focus blur. The parametric assumptions on blur kernels could greatly improve the robustness of blind image deblurring.

The out-of-focus blur occurs in the imaging process when the image plane is away from the ideal reference plane. The Gaussian and uniform disk functions [11] are frequently used in out-of-focus image deblurring. The Gaussian PSF is of the form

$$h_\sigma(x, y) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right), \tag{2}$$

where (x, y) are the pixel coordinates. The parameter σ, which is the standard deviation of the Gaussian function, can be estimated using the gradient information along edges [12], local frequency representation [13], or local contrast prior [14].

The disk PSF is defined as

$$h_r(x, y) = \begin{cases} \dfrac{1}{\pi r^2} & \text{if } \sqrt{x^2 + y^2} \leq r \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

The radius r can be estimated using circular Hough transform [15] or circular Radon transform [16]. We note that, although the Gaussian and disk kernels are widely employed, it has not been verified if they are the best parametric forms for out-of-focus blur modeling.

In this work, we propose that the out-of-focus blur kernel can be more appropriately depicted by the generalized Gaussian (GG) function. This is motivated by both theoretical analysis and practical observations. The optical theory reveals that the out-of-focus blur is the combined effect of lens defocusing and light diffraction. As a consequence, the blur kernel generally contains a flat hat and ramp edges that can be well modeled by the GG function. This is also verified on the blur kernels estimated from real images. Our observation further shows that the ramp edges of the blur kernels are of specific shapes, which implies that the two parameters α and β of the GG function are redundant in representing the blur kernel. Hence we establish the relationship between the two parameters by numerical simulation and obtain the

Manuscript received September 26, 2019; revised December 24, 2019 and March 27, 2020; accepted April 23, 2020. Date of publication April 27, 2020; date of current version March 5, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61371160 and in part by the Zhejiang Provincial Natural Science Foundation of China under Grant LY18F010007. This article was recommended by Associate Editor W. Liu. *(Corresponding author: Hui-Liang Shen.)*

Yu-Qi Liu, Xin Du, and Hui-Liang Shen are with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China (e-mail: liuyuqi@zju.edu.cn; duxin@zju.edu.cn; shenhl@zju.edu.cn).

Shu-Jie Chen is with the School of Computer and Information, Zhejiang Gongshang University, Hangzhou 310018, China (e-mail: chenshujie@zjgsu.edu.cn).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2020.2990623

simplified GG function with a single parameter β for blur kernel representation. We employ a strategy similar to [12] to compute the gradient ratio R_0 between the blurred and re-blurred step edges, and establish a look-up table (LUT) between R_0 and β. Finally, we compute the clear image using the obtained blur kernel and a non-blind image deblurring algorithm. In summary, the main contributions of this work are as follows:

- We propose that the GG function is more accurate in depicting the out-of-focus blur kernels compared with the conventional Gaussian and uniform disk models. This is supported by both theoretical and practical analysis.
- We simplify the two-parameter GG function to its single-parameter form based on the observation that the ramp edges of real blur kernels are of specific shapes. The parameter is robustly estimated using the step edges in the image.
- We show that our GG blur kernel estimation algorithm outperforms both parametric and nonparametric algorithms on both synthetic and real images, and is considerably beneficial to non-blind image deblurring.

A. Organization

The remainder of this paper is organized as follows. Section II briefly reviews the related work on parametric blur kernel estimation and image deblurring algorithms. Section III presents the theoretical and practical analysis of out-of-focus blur and argues that the blur kernel can be better modeled by the GG function. Section IV introduces the single-parameter GG model and elaborates the procedure of parameter estimation. Section V shows the experimental results, and finally Section VI concludes the paper.

II. Related Work

Image deblurring can be coarsely classified into two categories. One category is blind image deblurring, which usually employs an alternative framework to compute the latent image and blur kernel simultaneously [17]. Some natural image priors have been employed to reduce the ill-posed nature of this problem [7], [17]–[24]. The priors are usually discovered by analyzing the statistics of different types of blurred and sharp images. The other category is non-blind image deblurring. It either assumes a known blur kernel or estimates the blur kernel separately from deblurring algorithms [25]–[28]. In some circumstances such as out-of-focus blur and linear uniform motion blur, the blur kernels are of parametric forms whose parameters can be estimated from the blurred images. The readers can refer to [4] for a comprehensive overview of image deblurring. In the following, we only briefly review the previous work closely related to out-of-focus image deblurring.

A. Parametric Out-of-Focus Blur Kernels

As mentioned in Section I, the out-of-focus blur kernel is usually modeled as Gaussian or uniform disk functions. For example, the work [13] employs the Gaussian function to approximate the out-of-focus blur, and computes the scene depth using a pair of sharp and defocused images. However, in most cases only one blurred image is available. In [12], the defocus map is computed under the assumption that the out-of-focus blur kernel is Gaussian. The algorithm imposes an additional Gaussian blur on the original blurred image, and then estimates the Gaussian parameter using the gradient ratios from the original and re-blurred images along step edges. The work [29] presents an algorithm for detecting and estimating just noticeable blur (JNB) caused by defocusing that spans a small number of pixels in an image.

Approximating the out-of-focus blur by the Gaussian function is reasonable when the blur degree is small. However, it is noticed that, when the blur degree becomes large, the out-of-focus kernel has a flat top area [30] that cannot be well represented by the Gaussian function. In this regard, the out-of-focus blur kernel is also approximated as a uniform disk.

Early work on estimating the radius of uniform disk is introduced in [15]. It makes use of the circular Hough transform on the frequency domain. Later work adopts particle swarm optimization and wavelets transform to estimate the radius [31]. The work [16] presents an out-of-focus blur estimation algorithm using the circular Radon (C-Radon) transform. In the algorithm, a two-region power law function is introduced to fit the C-Radon curve. Unfortunately, the C-Radon transform would fail to characterize the out-of-focus blur for certain scenes because the frequency magnitudes are highly anisotropic.

Previous work on out-of-focus blur is mainly estimating the parameters of the Gaussian function and the disk kernel which are derived from the physical model and optic model respectively [11]. Both blur kernels are widely used for their simple representation and computation. The generalized Gaussian (GG) function has been deployed in approximating the blur kernel of atmospheric turbulence images [32].

In this work, we model the out-of-focus blur using the GG function in view of its accuracy. We simplify the parameters such that we can determine the GG blur kernel by a single parameter. We also propose an algorithm that estimates the single-parameter GG kernel from an out-of-focus blurred image based on our edge patch searching strategy.

B. Image Deblurring

Blind image deblurring refers to restoring the latent image from its blurred observation. With a known blur kernel the blind image deblurring is transformed to the non-blind image deblurring problem. Early non-blind image deblurring algorithms recover poor latent image with heavy ringing artifacts when the kernel estimation is not precise [18], [25]. Extensive image priors have been introduced to reduce the artifacts in the restored latent images where most of them are derived from the sparsity property of image gradients.

In the pioneering work [23] the heavy-tail distribution of gradients is approximated by concatenating a two piece-wise continuous functions as a prior. Inspired by the distribution of gradients, a fast deconvolution algorithm based on hyper-Laplacian function has been introduced in [17]. A LUT is used to speed up the deconvolution. Similar to [23], the work [20] observes the sparsity of gradients in text images and introduces an L_0 regularized prior based image deblurring algorithm. The L_0 regularized prior is also effective for natural images and the best among the state-of-the-art algorithms [33]. Modifications have been introduced to cope with saturated regions [21] where the L_0 regularization is imposed on the dark channel.

Over the last a few years, with the success of deep learning, approaches based on convolutional neural networks (CNNs) have been introduced. The CNN is trained to encode image features and learn the deconvolution operation without the need to know the cause of visual artifacts. In [26], a multi-layer perceptron (MLP) approach is developed to remove noise and artifacts which are produced by the deconvolution process. The work [27] adopts deep CNNs for non-blind image deconvolution and artifacts removal. In [34] a 6-layer CNN is used to restore sharp edges for blur kernel estimation. An end-to-end spatially variant recurrent neural networks (RNNs) is introduced in [28], whose weights are learned by a deep CNN for dynamic scene deblurring. Video deblurring techniques, which take advantage of the abundant information from neighboring frames for deblurring, also benefits from deep learning. In [35], an end-to-end CNN is employed to deblur an image given its neighboring frames. A 3D CNN is introduced in [36] to recover the sharp image with spatial-temporal features. A common drawback of deep learning algorithms is that they usually require large datasets for training and do not generalize well on unseen data.

III. Why Generalized Gaussian Model

The out-of-focus blur kernel is usually approximated by Gaussian or disk functions in many image deblurring methods [13], [37], [38]. This section shows that the GG function is more accurate in depicting the out-of-focus blur based on theoretical analysis and practical observation.

A. Optical Theory of Out-of-Focus Blur

The out-of-focus blur occurs when the distance between the image plane and the sensor reference plane is large enough to produce an unacceptable defocus blur that adversely affects the performance of the optical system [39]. Assuming a point of light source on the focal plane whose distance between the lens is S_1, the light will focus on a spot on the sensor reference plane. If we place the light source farther where the distance to lens becomes S_2, the rays of light will converge before the sensor reference plane. As a result, a blurred pattern will appear on the sensor. The pattern is a scaled version of the aperture, for example, a uniform disk which is called the CoC [19]. Fig. 1(a) illustrates the optical out-of-focus model with aperture diameter A and focal length f. The diameter c of the CoC is formulated as

$$c = \frac{|S_2 - S_1|}{S_2} \cdot \frac{fA}{S_1 - f}. \tag{4}$$

The diffraction is inevitable even in a perfect optical system when out-of-focus blur occurs. Similarly, we consider an ideal small point source of light on the focal plane captured by the optical system. The light will show as a diffraction pattern on the screen called Airy disk rather than a bright point [40]. Fig. 1(b) illustrates the diffraction pattern and its optical model. The intensity of the diffraction on angle θ is given by [39]

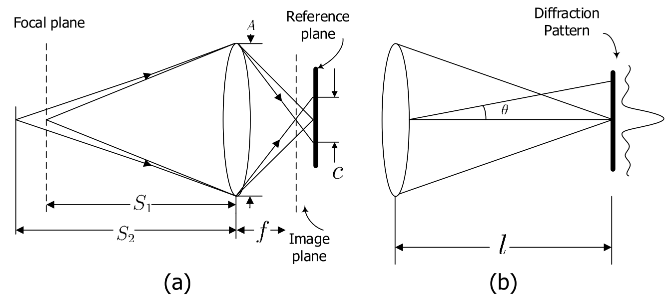

Fig. 1. Two optical models closely related to out-of-focus blur. (a) Out-of-focus blur. (b) Optical diffraction.

$$I(\theta) = I_0 \cdot \left(\frac{2J_1(m)}{m}\right)^2, \tag{5}$$

where I_0 denotes the maximum intensity of the pattern at the center of Airy disk pattern, J_1 is the Bessel function of the first kind of order one, and

$$m = \frac{\pi A \sin\theta}{\lambda}, \tag{6}$$

where λ denotes the wavelength of the light source.

The reason for the diffraction pattern is that light travels as a wave motion and thus it bends around corners. The observed out-of-focus blur is the combined output of the out-of-focus model and the diffraction model. Hence, the out-of-focus blur kernel can be represented by the convolution of a uniform disk (CoC) with the Airy disk because every single ray of light suffers from the diffraction. Fig. 2 shows the 2D PSFs of the disk kernel, Airy disk, and their convolution as well as the corresponding profiles. It is obvious that the shape of the convolution is much like a GG distribution than the Gaussian and the disk kernel. This is also supported in the practical observation as will be presented in the following.

B. Real Out-of-Focus Blur Kernel

Previous work has shown that the blur kernel is a scaled version of the aperture [19]. To approximate circular apertures, the normal lens uses 5-bladed, 6-bladed and 7-bladed diaphragms. We estimate the out-of-focus blur kernels using 7-bladed apertures with a tripod and find that the blur kernels are isotropic. The isotropy is possibly due to the diffraction and the increment of the blade number. The real out-of-focus blur kernel is estimated through a maximum a posteriori (MAP) algorithm using a sparse prior [41]. Fig. 3 illustrates the apertures of different f-numbers and the calibration chart. The chart contains circles of different sizes such that the richness of gradient directions can benefit blur kernel estimation. Then we acquire a set of blurred images $\mathbf{f}$ and sharp images $\mathbf{g}$ of the calibration chart. An alignment algorithm is used to reduce

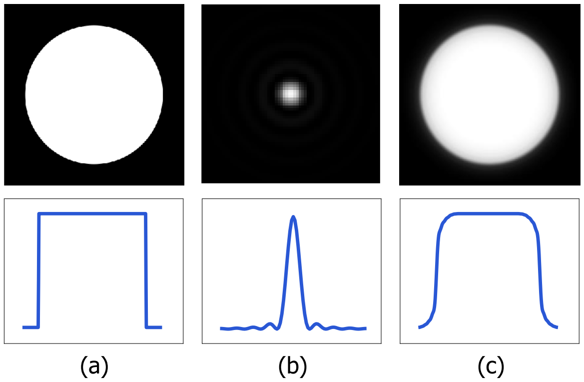

Fig. 2. The CoC, Airy disk and their convolution. (a) The 2D PSF of the uniform disk-shaped CoC and its profile. (b) The 2D Airy disk pattern and its profile. (c) The 2D PSF of the convolution of CoC and Airy disk and its profile.

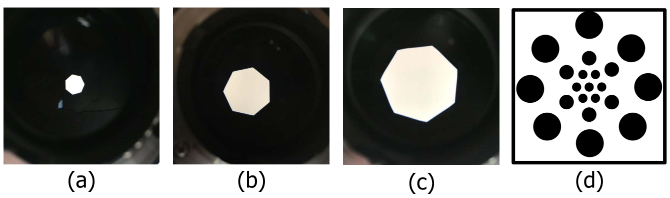

Fig. 3. The components used for blur kernel estimation. (a)-(c) A heptagon lens with aperture partially closed. (d) The calibration chart.

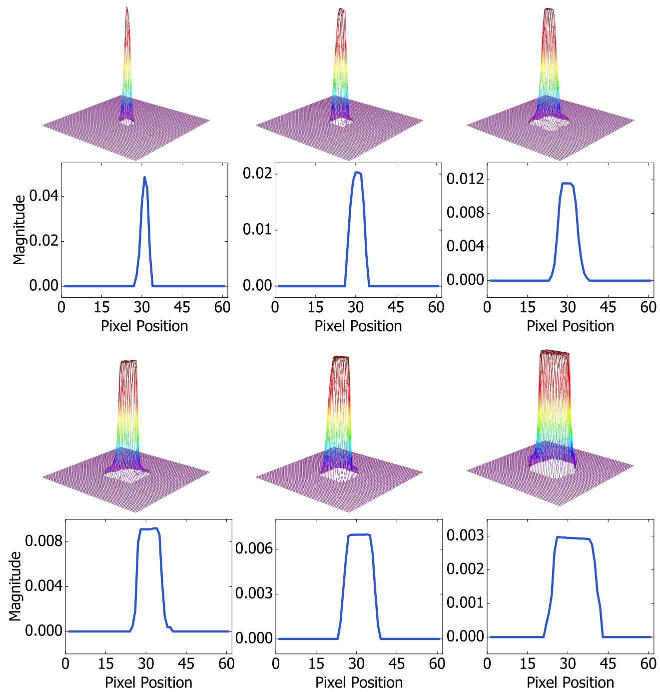

Fig. 4. The blur kernel estimated from the blurred and sharp images of the calibration chart.

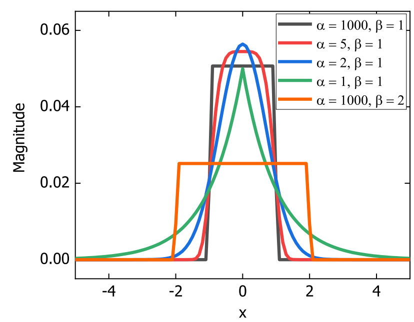

Fig. 5. The shapes of one-dimensional GG functions generated using different parameter values.

the displacement between the blurred and the sharp image [42] to improve the kernel estimation accuracy. The blur kernel is computed by minimizing the objective function

$$J(\mathbf{h}) = \|\mathbf{g} * \mathbf{h} - \mathbf{f}\|_2^2 + \mu_1 \cdot \phi(\mathbf{h}), \tag{7}$$

where μ_1 balances the weight of the data term and regularization, and is set to 0.002 in this work. The constraint $\phi(\mathbf{h})$ is defined as

$$\phi(\mathbf{h}) = \sum_i \phi(h_i), \quad \phi(h_i) = \begin{cases} h_i, & h_i \geq 0, \\ +\infty, & h_i < 0 \end{cases} \tag{8}$$

where h_i denotes the ith entry in the blur kernel.

Several estimated 2D blur kernels and their profiles along the x-axis are shown in Fig. 4. It is observed that the kernels have ramp edges (instead of step edges) and top flat areas, and the decay rates of the edges are similar among different kernels. Based on this observation, the disk and Gaussian functions are inaccurate in modeling the out-of-focus blur kernel. This inspires us to seek a more suitable kernel form.

C. Modeling Blur Kernels Using the GG Function

The GG function is capable of describing the shapes of the kernels illustrated in Fig. 4. It has been formerly applied in the computer vision fields such as the atmospheric turbulence [32]. The GG blur kernel involves two parameters α and β, which control the decay rate of edges and the scale respectively. To keep the image intensity level unchanged, the GG blur kernels are normalized so that they sum to 1. The 2D GG function is represented as

$$G_{\alpha,\beta}(x, y) = \frac{\alpha}{2\pi\beta^2\ \Gamma(\frac{2}{\alpha})} \exp\left(-\left(\frac{\sqrt{x^2+y^2}}{\beta}\right)^{\alpha}\right), \tag{9}$$

where $\Gamma(\cdot)$ is the Gamma function. Fig. 5 shows the GG functions generated with different α and β values. As observed, the GG function becomes the Laplacian, Gaussian, and disk functions when $\alpha = 1, 2,$ or ∞.

IV. Estimating the GG Blur Kernels

In this section, we first show that the 2D GG function can be simplified to a 1D function based on the characteristics of the out-of-focus blur kernels. Then we estimate the parameter of the GG blur kernel using image patches containing step edges. Finally, we present the non-blind image deblurring algorithm using the estimated blur kernel.

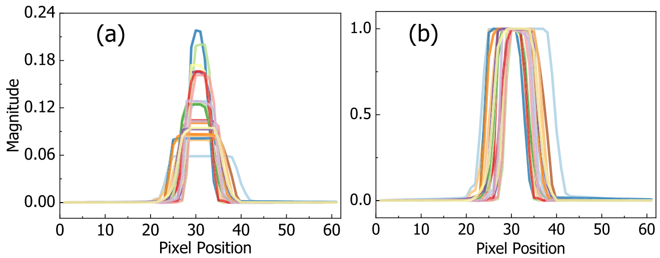

Fig. 6. The profiles of 24 estimated out-of-focus blur kernels. (a) Original profiles. (b) Scaled profiles.

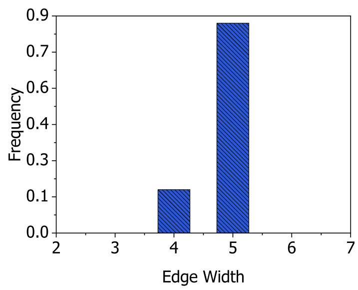

Fig. 7. The distribution of the width of the falling and rising edges computed from the profiles of real kernels.

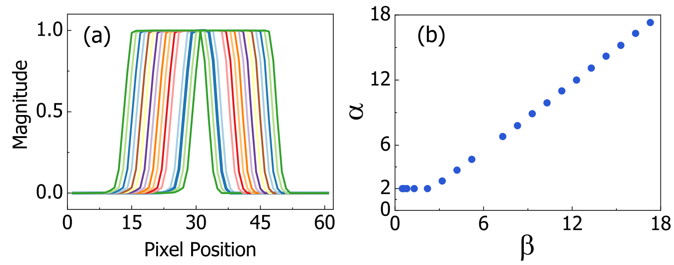

Fig. 8. Computation of the relationship between the parameters α and β. (a) Synthetic blur kernels. (b) The distributions of the optimal parameters.

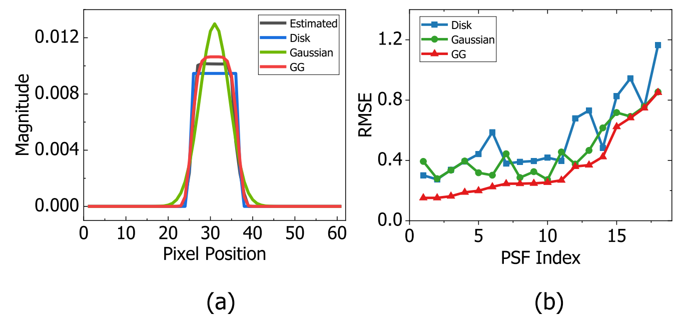

Fig. 9. The comparison of different kernels. (a) The estimated real kernel and the fitting results. (b) RMSEs of different kernels.

A. Simplifying the GG Model

For observation, we plot the profiles of 24 out-of-focus blur kernels in Fig. 6(a). As the entries of a kernel sum to 1, the magnitudes of these profiles are different. To further explore the characteristics of these kernels, we scale these profiles such that their maximum magnitudes become 1. Fig. 6(b) reveals that the width of the raising and falling edges are very similar for all kernels.

To quantify the similarity, we define the edge width as the amount of pixels between two adjacent positions with maximum curvatures. Let $i(x)$ be the magnitude of the profile, the curvature at position x is computed as

$$k(x) = \frac{|i''(x)|}{\left(1 + (i'(x))^2\right)^{\frac{3}{2}}}, \tag{10}$$

where $i'(x)$ and $i''(x)$ denote the first and second derivatives of $i(x)$, respectively. We plot the histogram of the 48 edge width (24 for raising edges and 24 for falling edges) in Fig. 7, which indicates that most edges are of 5-pixel widths.

This observation inspires us that the two parameters, α and β, of the GG function are redundant in depicting the out-of-focus blur kernel. More specifically, these two parameters should keep a relationship such that the GG function can produce blur kernels with a 5-pixel edge width. Because it is difficult to formulate the relationship between α and β directly, we explore this using numerical analysis. To exclude the influence of noise, we generate a set of ideal blur kernels as shown in Fig. 8(a) by simulating them from a real blur kernel. As shown, these kernels have identical edge widths but different spans between the rising and falling edges. We then normalize the blur kernel such that the kernel entries sum to 1.

We determine the relationship between α and β using exhaustive search under the constraint that the blur kernels should keep a constant edge width. The parameter α should be in the range $[2, \infty)$. This is due to the fact that, even when the object is well-focused, the acquired image is still blurred by the Airy disk. As most energy is inside the first bright ring, the Airy disk can be approximated as a Gaussian function [39], which corresponds to $\alpha = 2$. We compute the optimal parameters for the simulated kernels in Fig. 8(a) in terms of lowest root mean squared error (RMSE), and plot the optimal parameter values in Fig. 8(b). We observe that α and β have a linear relationship when $\alpha > 2$. More specifically, α can be expressed as a function of β,

$$\begin{aligned} \alpha &\equiv f(\beta) \\ &= \max(2,\ 1.033\beta - 0.6217). \end{aligned} \tag{11}$$

In this manner, the two-parameter GG blur kernel can be simplified to its single-parameter form,

$$G_\beta(x, y) = G_{\alpha,\beta}(x, y)|_{\alpha = f(\beta)}, \tag{12}$$

which can improve robustness in parameter estimation.

For quantitative analysis, we compare the capability of the Gaussian, disk, and single-parameter GG functions in depicting real out-of-focus blur kernels. Fig. 9(a) shows the fitting results of a typical real blur kernel, and Fig. 9(b) illustrates the fitting RMSEs of different kernels. It is clear that our single-parameter GG function is more appropriate in depicting out-of-focus blur kernels. This is in accordance with the optical theory and the practical observation in Section III.

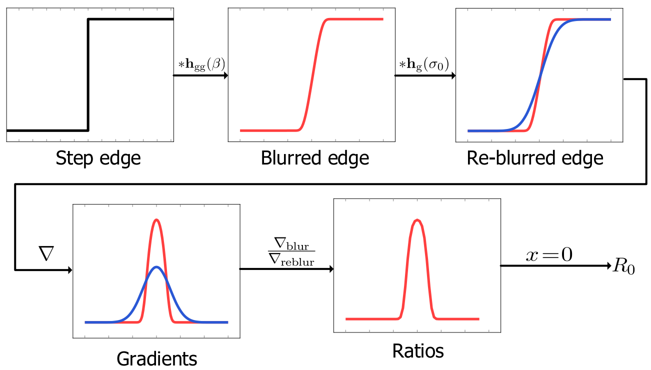

Fig. 10. The flowchart of parameter estimation. Here, $\mathbf{h}$ denotes the out-of-focus blur kernel, $\mathbf{h}_g(\sigma_0)$ denotes the Gaussian function with specified standard deviation σ_0, and ∇ denotes the gradient operator.

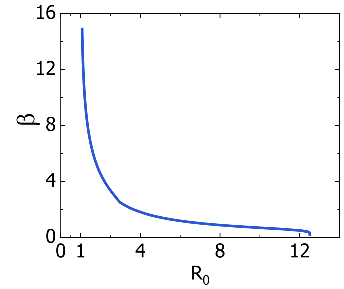

Fig. 11. The shape of $LUT(R_0, \beta)$.

B. Parameter Estimation

Our parameter estimation algorithm is inspired by the gradient-based framework introduced in [12]. Fig. 10 illustrates the flow chart of the algorithm, which consists of several steps. The ideal step edge is corrupted by the out-of-focus blur kernel $\mathbf{h}$ which can be modeled as the single-parameter GG function. We re-blur the blurred edge with a Gaussian function $\mathbf{h}_g(\sigma_0)$ with known standard deviation σ_0 to generate the re-blurred edge. Then, we compute the ratio distribution of the gradients of the blurred and the re-blurred edges. Finally, we use the ratio R_0 at the position $x = 0$ to further compute the GG parameter β.

We note that the out-of-focus blur kernel is approximated as a Gaussian function in [12], which takes advantage of the property that the convolution of two Gaussian functions is still Gaussian. Hence the variance of the unknown Gaussian blur kernel can be computed from the gradient ratio using a simple formula. Although this formula does not hold when using the GG kernel function, we will show that the GG parameter β can be determined from the ratio R_0 by establishing a LUT, denoted as $LUT(R_0, \beta)$.

In the following, we elaborate the parameter estimation algorithm based on our single-parameter GG kernel. The 2D step edge is denoted by $\mathbf{s} = a\mathbf{u}_x + b$, where $\mathbf{u}_x$ is a 2D step function whose magnitude changes at $x = 0$. The out-of-focus blur on the step edge is modeled as

$$\mathbf{t} = \mathbf{h}_{gg}(\beta) * \mathbf{s}, \tag{13}$$

where $*$ denotes the convolution operator. After consequent Gaussian re-blurring, the edge further becomes

$$\mathbf{t}_1 = \mathbf{h}_{gg}(\beta) * \mathbf{s} * \mathbf{h}_g(\sigma_0). \tag{14}$$

In this work, we set $\sigma_0 = 5$. Then the gradient ratio can be computed at each pixel position,

$$R = \frac{\nabla \mathbf{t}}{\nabla \mathbf{t}_1} = \frac{\nabla\left(\mathbf{h}_{gg}(\beta) * \mathbf{s}\right)}{\nabla\left(\mathbf{h}_{gg}(\beta) * \mathbf{s} * \mathbf{h}_g(\sigma_0)\right)}, \tag{15}$$

where ∇ denotes the gradient operator.

We take the ratio R_0, which locates at the edge position $x = 0$, as the representative of the ratio distribution R. Although R_0 may not be the maximum when employing the GG kernel function, its magnitude is comparatively large in the ratio distribution. This is beneficial to a robust parameter estimation.

By assuming the edge locates on the y-axis, (15) could be further simplified. Note that $\nabla\mathbf{t}$ and $\nabla\mathbf{t}_1$ are symmetry to y-axis because the GG and Gaussian functions are both centrally symmetric. Then R_0 can be represented as

$$R_0 = \frac{\nabla t(0,0)}{\nabla t_1(0,0)} = \frac{\int_{-\infty}^{\infty} h_{gg}(\beta, 0, y)dy}{\int_{-\infty}^{\infty} h(\beta, \sigma_0, 0, y)dy}, \tag{16}$$

where $t(0,0)$ and $t_1(0,0)$ denote the edge magnitudes at the origin, $h_{gg}(\beta, 0, y)$ denotes the magnitudes along the y-axis of GG kernel, and

$$h(\beta, \sigma_0, x, y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h_{gg}(\beta, x-u, y-v)h_g(\sigma_0, x, y)dudv. \tag{17}$$

Here $h_g(\sigma_0, x, y)$ denotes the magnitude of the re-blurring Gaussian function at position (x, y).

Note that our objective is to determine the GG parameter β from the estimated gradient ratio R_0. As the analytical form does not exist, we instead establish the numerical $LUT(R_0, \beta)$ according to expression (16). The shape of the LUT is illustrated in Fig. 11.

C. Locating Edge Patches

The above analysis indicates that, under the assumption of step edge, computing the GG parameter β is deterministic after obtaining the gradient ratio R_0. Hence locating step edges is important for parameter estimation. The previous work [12] detects edge patches using the Canny detector [43]. Its limitation is that the detected patches may contain curved and irregular edges that violate the step-edge assumption.

In this work we locate edge patches using template matching, evaluated in terms of the normalized cross-correlation (NCC) measure [44]. As our aim is to locate appropriate edges but not to detect all edges, we only employ 4 templates with 0°, 45°, 90°, and 135° edge directions. The patch size is set to 25 × 25 and the NCC threshold is set to 0.9. Fig. 12 shows an example for edge patch detection. It is observed our approach performs better than the Canny edge detector in locating step-like edges. It is common that the detected patches

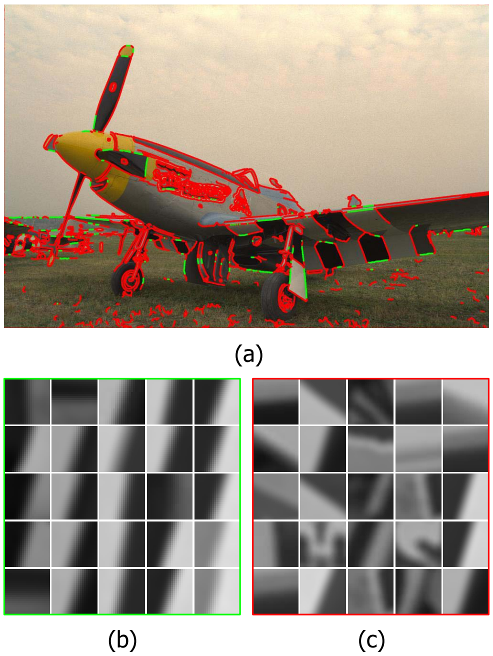

Fig. 12. Edge patch detection. (a) The blurred image. The overlapped red and green points denotes the centers of the patches obtained by our approach and the Canny detector, respectively. (b) Edge patches detected by our approach. (c) Edge patches detected by the Canny detector.

Algorithm 1 Computing GG Parameter β

Initialization: Prepare $LUT(R_0, \beta)$ according to (16).
Input: The out-of-focus blur image.
Output: GG parameter β.
1. Re-blur the input image using the Gaussian function $\mathbf{h}_g(\sigma_0)$;
2. Detect edge patches by template matching;
3. Compute the gradient ratio R_0 for each edge patch;
4. Determine the most possible R_0 by voting;
5. Determine β using $LUT(R_0, \beta)$.

produce different gradient ratios. By assuming that the out-of-focus is spatially uniform in the image, we determine the most possible gradient ratio R_0 by voting.

After edge patch detection and edge ratio computation, the GG parameter β can be computed according to Algorithm 1. Then we can construct the blur kernel using the obtained β. Our experiment reveals that the kernel size ($K \times K$) is not influential to the image deblurring accuracy provided that it covers 95% or above energy of the analytic GG model. Hence we set the kernel width as $K = 2 \times \lceil 2\beta \rceil + 1$, where the operator $\lceil \cdot \rceil$ rounds the variable up towards the nearest integer.

D. Image Deblurring

After obtaining the GG blur kernel, we employ the non-blind deblurring algorithm introduced in [20] to compute the sharp image. The objective function is defined as

$$\mathbf{g}_0 = \arg\min_{\mathbf{g}} \|\mathbf{g} * \mathbf{h} - \mathbf{f}\|_2^2 + \mu_2 \|\nabla \mathbf{g}\|_0, \qquad (18)$$

where $\mathbf{g}_0$, $\mathbf{h}$, $\mathbf{f}$ denote the latent image, the blur kernel, and blur observation respectively. $\|\cdot\|_0$ denotes the L_0 norm. The parameter μ_2 balances the weight between the data term and gradient regularization, and is set to 0.5 in this work. We use the ringing suppression algorithm [20] to remove artifacts, which can be presented as

$$\mathbf{g} = \mathbf{g}_l - \mu_3 \cdot BF(\mathbf{g}_l - \mathbf{g}_0), \qquad (19)$$

where BF denotes the bilateral filter, $\mathbf{g}_l$ is the latent image estimated with Laplacian prior in [17]. μ_3 controls the degree of ringing suppression and is set to 0.5 in this work. The L_0 prior based deblurring algorithm [20] was introduced for text image deblurring but also works effectively on natural images.

V. Experiments

We evaluate the accuracy of our single-parameter GG kernel estimation algorithm on both synthetic and real images. The synthetic blurred images are generated by convolving various GG blur kernels with sharp images collected from the Kodak [45], BSD500 [46], and COCO [47] datasets, and from the Internet. The real scene images are acquired by ourselves using a single lens reflex (SLR) camera.

We compare our kernel estimation algorithm with the parametric algorithms, Zhuo and Sim [12] and Oliveira *et al.* [16], which approximate the out-of-focus blur kernel as Gaussian functions and disk functions, respectively. The width of the square Gaussian kernel for Zhuo and Sim [12] is set as $K_1 = 2 \times \lceil 3\sigma \rceil + 1$. We also compare our algorithm with the state-of-the-art blind deblurring algorithms including Pan *et al.* [20] and Yan *et al.* [22]. Both algorithms [20], [22] employ the coarse-to-fine strategy to estimate the square-sized blur kernel. The kernel width is set as $K_2 = \lceil 2\beta \rceil + 3$, where β denotes the ground truth (in synthetic data experiments) or estimated (in real data experiments) parameter of the GG kernel. The kernel of the coarsest level is initialized as Gaussian with standard deviation $\sigma_0 = 0.5$. Our experiment shows that these settings are proper to the algorithms [20], [22].

Two deep learning based deblurring algorithms [28], [34] are used for comparison as well. Xu *et al.* [34] estimates the blur kernel, while Zhang *et al.* [28] restores the sharp image without producing the blur kernel. For a fair comparison of the blur kernels produced by different algorithms, we employ the same non-blind deconvolution framework elaborated in Section IV-D.

We conduct quantitative evaluation on synthetic images. The accuracy of blur kernel is evaluated using the RMSE computed between the estimated and ground truth kernels. The deblurring quality is evaluated by the peak signal-to-noise ratio (PSNR) and structure similarity (SSIM) [48] measures computed between the deblurred and ground truth sharp images. We also conduct qualitative evaluation on real scene images, by judging the visual quality of the estimated blur kernels and deblurred images.

A. Image Datasets

Fig. 13 shows the sample images collected from the Kodak dataset [45] and from the Internet. The images *Wall* and

Fig. 13. Images from the Kodak dataset [45] and the Internet.

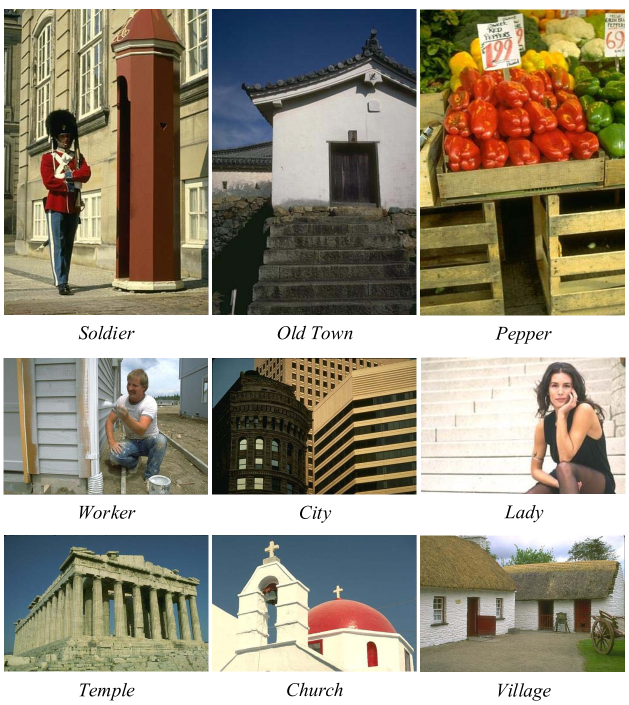

Fig. 14. Images from the BSD500 dataset [46].

Bookshelf are indoor scenes while the others are outdoor scenes. The image sizes are larger than 10^6 pixels.

Fig. 14 shows the images selected from the BSD500 dataset [46] that includes the buildings, portraits, and vegetables scenes. The image sizes are about 2.5×10^5 pixels, much smaller than those of the Kodak dataset. Fig. 15 shows the images collected from the COCO dataset [47].

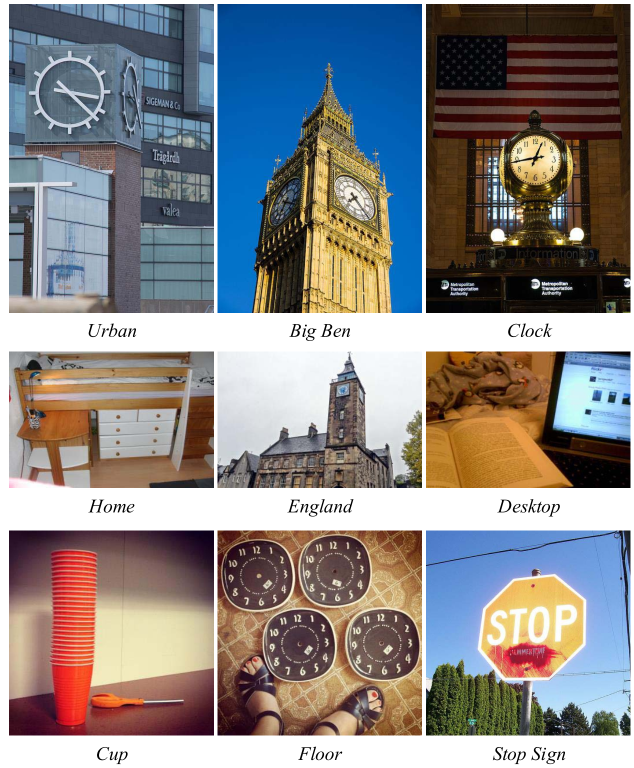

Fig. 15. Images from the COCO dataset [47].

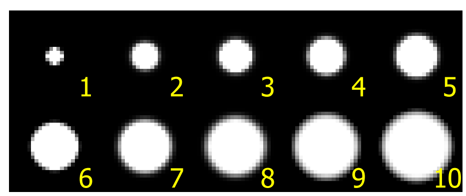

Fig. 16. The 10 single-parameter GG blur kernels used for simulating out-of-focus blur images. The parameter $\beta = 1, 2, \cdots, 10$.

They are mostly human-made scenes and the image sizes are also small.

B. Accuracy of Blur Kernel Estimation

The accuracy of the blur kernel estimation is fundamental to non-blind image deblurring since an inappropriate blur kernel will produce ringing artifacts. We evaluate the accuracy on simulated blurred images by using various GG blur kernels to generate the synthetic images (see Fig. 16). The accuracy of kernel estimation is evaluated using RMSE, which is computed as

$$RMSE = \sqrt{\frac{1}{N}\|\mathbf{h} - \mathbf{h}_{\mathrm{gt}}\|_2^2}, \tag{20}$$

where N denotes the number of entries in the blur kernel, $\mathbf{h}$ and $\mathbf{h}_{\mathrm{gt}}$ denote the estimated and ground truth kernels, respectively.

Fig. 17 shows the average RMSEs of different blur kernel estimation algorithms. It is observed that the RMSEs produced by our algorithm is close to those of Zhuo and Sim [12]

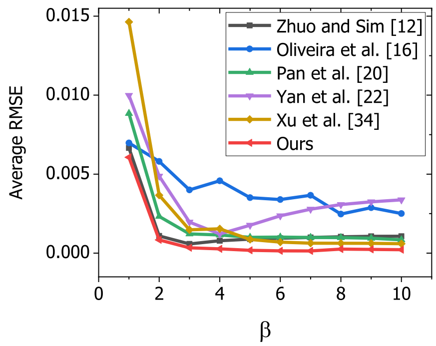

Fig. 17. RMSE values of the blur kernels estimated by different algorithms.

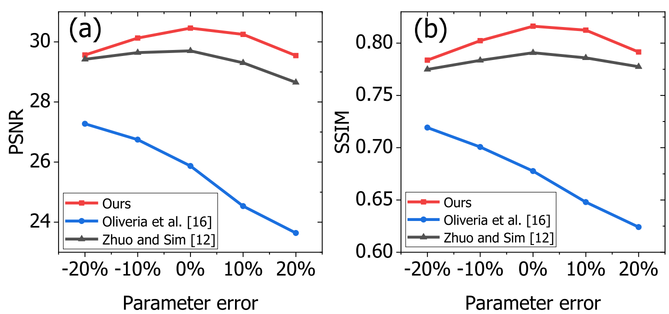

Fig. 18. The average PSNR and SSIM values produced by the three deblurring algorithms in case of parameter estimation biases.

when $\beta \leq 2$. This is because that the single-parameter GG kernel approaches Gaussian when β is small.

We also evaluated the robustness of image deblurring by assuming that the kernel parameters are biased by $\pm 10\%$ and $\pm 20\%$. Fig. 18 shows the average PSNR and SSIM values of the deblurring results using the kernels estimated by Zhuo and Sim [12], Oliveira *et al.* [16], and our algorithm. It is observed that our algorithm slightly degrades when inducing parameter biases but still performs better than the two competitors. Interestingly, the PSNR and SSIM values of Oliveira *et al.* [16] become larger in the cases of -10% and -20% parameter biases. This indicates that the algorithm is not quite robust in kernel estimation.

C. Deblurring Results on Synthetic Images

After obtaining the parametric or nonparametric out-of-focus blur kernel, we compute the sharp images using the same non-blind image deblurring procedure for a fair comparison [20]. Table I, Table II and Table III list the average PSNR and SSIM values of different deblurring algorithms, which are computed on each scene blurred by the 10 GG blur kernels illustrated in Fig. 16. It is observed that our algorithm produces the highest or the second highest PSNR and SSIM values on most synthetic scenes. Zhang *et al.* [28] also gives good accuracy, indicating that an end-to-end network is effective in image deblurring. Compared with Zhang *et al.* [28], the advantage of our algorithm is that it is very simple and does not require a training dataset.

In addition to the quantitative comparison, we also illustrate the image deblurring results on synthetic scenes for visual comparison. Fig. 19 shows the estimated blur kernels produced by different algorithms and their corresponding deblurring results on the *Villa* scene. It is observed that the deblurring results produced by Oliveria *et al.* [16] contains obvious ringing artifacts. The error maps indicate that, our algorithm, Zhuo and Sim [12], and Zhang *et al.* [28] perform better than the three nonparametric algorithms, Pan *et al.* [20], Yan *et al.* [22], and Xu *et al.* [34].

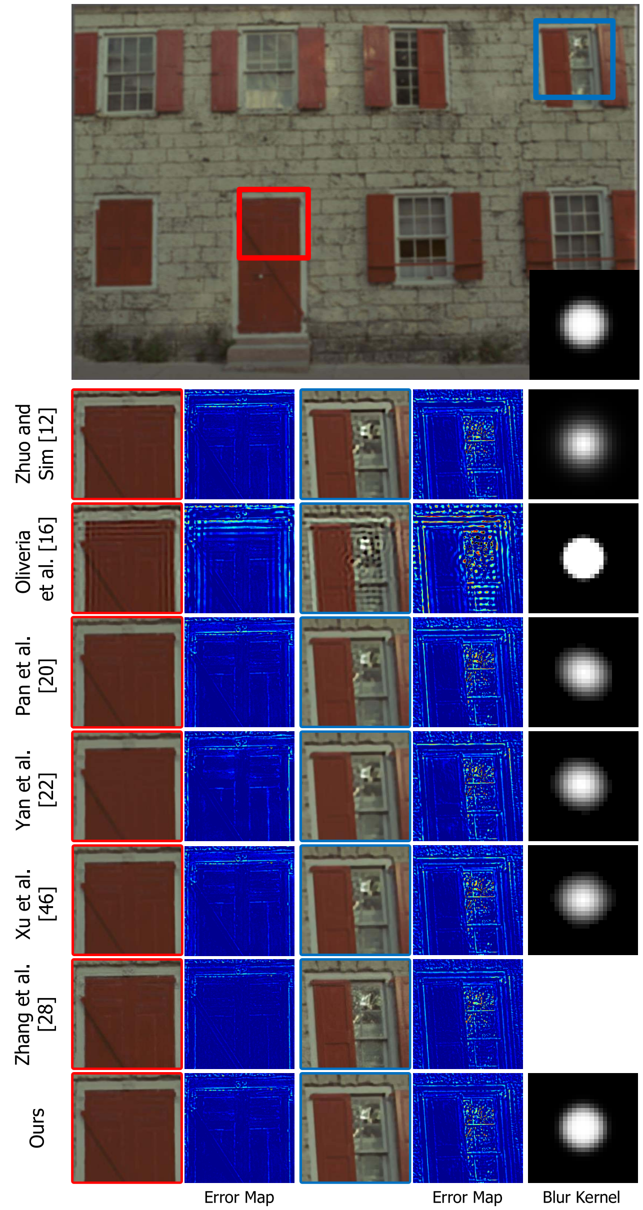

Fig. 19. The deblurring results on the synthetic image *Villa* blurred by the GG kernel with parameter $\beta = 5$. Top: Original image and the ground truth blur kernel. Bottom: Highlighted deblurring results and their error maps computed w.r.t. the ground truth sharp images (1st - 4th columns) and the estimated blur kernels (5th column). Note that the kernels produced by individual algorithms have been resized for visualization.

Fig. 20 shows that the Gaussian kernel estimated by Zhuo and Sim [12] is quite different from the ground truth kernel, and the restored image is still blurry. As the GG kernel cannot be approximated by a uniform disk, Oliveria *et al.*'s algorithm

TABLE I

PSNR AND SSIM OF THE DEBLURRING RESULTS COMPUTED BY DIFFERENT ALGORITHMS ON THE KODAK DATASET AND INTERNET IMAGES. THE BEST AND SECOND BEST ONES ARE IN RED AND BLUE, RESPECTIVELY

	Oliveira *et al.* [16]		Zhuo and Sim [12]		Pan *et al.* [20]		Yan *et al.* [22]		Xu *et al.* [34]		Zhang *et al.* [28]		Ours	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Villa	28.41	0.71	31.00	0.80	31.89	0.84	31.70	0.84	31.27	0.82	**32.16**	**0.84**	**32.22**	**0.85**
Sea	28.62	0.73	28.83	0.75	28.96	0.77	28.83	0.76	29.10	0.77	**30.92**	**0.82**	**29.73**	**0.78**
Sailboat2	32.68	0.73	31.54	0.87	**33.96**	0.90	33.85	0.90	33.00	0.89	**35.14**	**0.89**	32.90	**0.90**
Sailboat1	32.02	0.88	31.25	0.87	**33.39**	**0.90**	33.34	**0.90**	32.79	0.89	**35.33**	0.89	32.02	0.89
Lighthouse1	29.08	0.80	28.50	0.78	**30.30**	**0.83**	**30.30**	**0.83**	30.26	**0.83**	29.89	**0.83**	29.56	0.81
Lighthouse2	23.86	0.72	30.96	**0.89**	30.85	0.84	30.87	0.84	30.50	0.84	**31.72**	0.85	**31.85**	**0.91**
Bookshelf	11.37	0.37	23.09	0.75	23.64	0.79	23.43	0.78	22.37	0.75	**24.68**	**0.83**	**23.79**	**0.80**
Wall	27.45	0.95	30.81	0.98	31.97	0.98	31.97	0.98	31.45	0.98	**32.35**	**0.99**	**32.50**	**0.99**
Street	23.08	0.65	23.20	0.67	24.14	0.83	24.15	0.82	21.47	**0.75**	**24.35**	**0.84**	**24.77**	0.74

TABLE II

PSNR AND SSIM OF THE DEBLURRING RESULTS COMPUTED BY DIFFERENT ALGORITHMS ON THE BSD500 DATASET. THE BEST AND SECOND BEST ONES ARE IN RED AND BLUE, RESPECTIVELY

	Oliveira *et al.* [16]		Zhuo and Sim [12]		Pan *et al.* [20]		Yan *et al.* [22]		Xu *et al.* [34]		Zhang *et al.* [28]		Ours	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Soldier	17.80	0.67	21.53	0.78	22.24	0.82	22.23	0.82	21.21	0.81	**22.47**	**0.84**	**23.08**	**0.84**
Old Town	24.84	0.71	26.12	0.77	27.15	0.79	27.11	0.78	26.34	0.78	**27.79**	**0.82**	**27.71**	**0.80**
Pepper	21.96	0.86	23.46	0.89	24.22	0.92	24.35	0.92	23.88	0.92	**24.93**	**0.93**	**25.34**	**0.93**
Worker	22.54	0.68	25.02	0.76	24.91	0.77	25.02	0.77	24.11	0.76	**25.94**	**0.83**	**25.88**	**0.80**
City	22.06	0.72	22.75	0.75	22.85	0.77	23.26	0.79	21.96	0.77	**22.47**	**0.80**	**23.26**	**0.77**
Lady	21.94	0.73	23.82	0.77	22.99	0.77	23.03	0.77	22.33	0.76	**25.74**	**0.85**	**24.23**	**0.82**
Temple	23.83	0.83	24.48	0.83	24.71	0.85	24.80	0.85	24.15	0.85	**24.98**	**0.85**	**25.33**	**0.86**
Church	22.00	0.81	22.37	0.87	23.52	0.89	23.59	0.90	23.44	0.89	**24.88**	**0.92**	**24.71**	**0.91**
Village	19.69	0.60	22.46	0.72	22.93	0.73	23.00	0.73	22.88	0.73	**23.94**	**0.78**	**23.19**	**0.74**

TABLE III

PSNR AND SSIM OF THE DEBLURRING RESULTS COMPUTED BY DIFFERENT ALGORITHMS ON THE COCO DATASET. THE BEST AND THE SECOND BEST ONES IN RED AND BLUE, RESPECTIVELY

	Oliveira *et al.* [16]		Zhuo and Sim [12]		Pan *et al.* [20]		Yan *et al.* [22]		Xu *et al.* [34]		Zhang *et al.* [28]		Ours	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Urban	21.07	0.69	24.86	0.81	25.25	0.84	25.14	0.84	23.94	0.81	**25.70**	**0.87**	**26.10**	**0.85**
Big Ben	17.07	0.77	20.09	0.85	**21.63**	**0.87**	**21.63**	**0.87**	20.70	0.86	21.48	**0.88**	21.55	0.85
Clock	20.87	0.76	23.42	0.83	24.43	0.85	24.23	0.84	23.87	0.83	**24.21**	**0.86**	**24.74**	**0.85**
Home	24.59	0.77	26.08	0.84	26.97	0.87	26.96	0.87	26.51	0.87	**28.05**	**0.89**	**27.34**	**0.87**
England	20.89	0.72	20.50	0.72	21.08	0.73	20.88	**0.74**	20.90	0.73	**21.53**	**0.76**	**21.28**	0.72
Desktop	26.12	0.91	25.58	0.92	26.56	0.92	26.47	0.92	**26.90**	0.93	**28.08**	**0.95**	25.58	**0.94**
Cup	27.71	0.97	26.95	0.96	28.72	0.97	29.00	0.98	28.10	0.97	**29.27**	**0.98**	**31.12**	**0.98**
Floor	20.35	0.80	21.06	0.81	23.03	0.86	22.97	0.86	21.94	0.85	**23.37**	**0.89**	**23.38**	**0.86**
Stop Sign	21.10	0.81	21.17	0.84	**22.50**	0.85	22.49	0.85	21.72	0.84	**22.60**	**0.86**	22.40	**0.86**

produces severe ringing artifacts. The three nonparametric algorithms, Pan *et al.* [20]. Yan *et al.* [22], and Xu *et al.* [34] produce blurry results due to the inaccurate estimation of blur kernels. Zhang *et al.* [28] successfully recovers a sharp image, but still generates some artifacts near the text in the red box. In comparison, our estimated kernel is quite similar to the ground truth, and the restored image is relatively sharp and free of artifacts.

Fig. 21 shows that the deblurring results produced by Pan *et al.* [20] and Oliveria *et al.* [16] exhibit ringing artifacts around the stars and text due to inaccurate estimation of blur kernels. Zhuo and Sim [12], Yan *et al.* [22], and Xu *et al.* [34] cannot restore the text well in the deblurred images. Zhang *et al.* [28] produces a sharp image with noticeable ripple effects around the stars in the red box. In comparison, our algorithm generates a less sharp but visually pleasant deblurred image.

D. Deblurring Results on Real Images

We further evaluate the blur kernel estimation algorithms on real out-of-focus images whose size is around

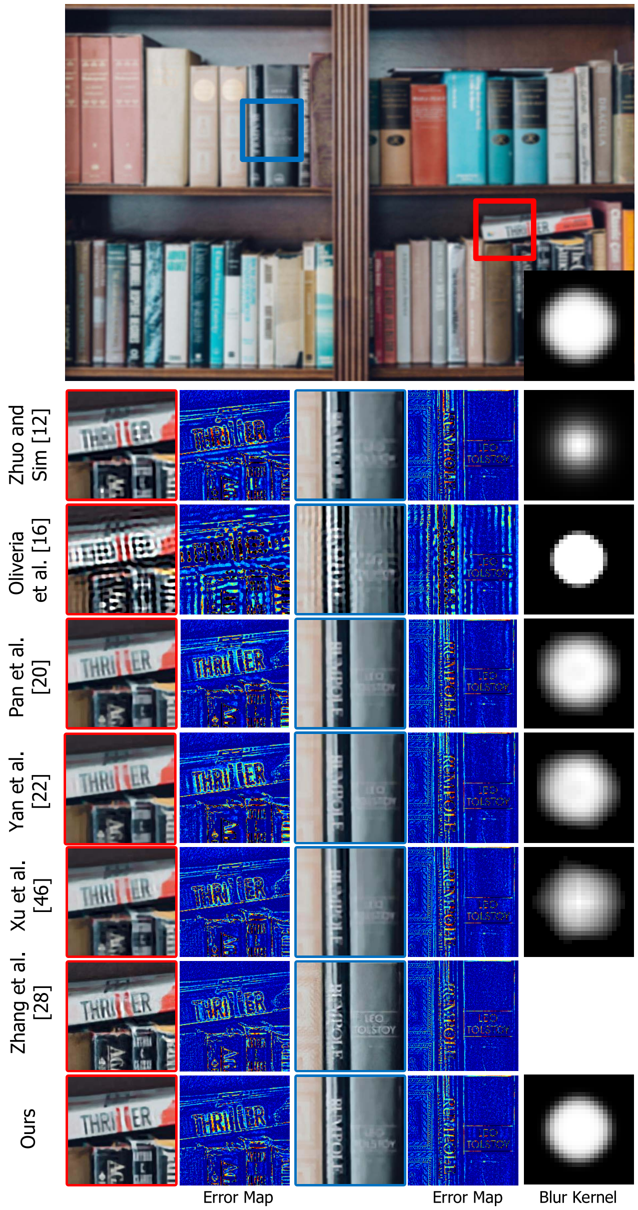

Fig. 20. The deblurring results on the synthetic images *Bookshelf* blurred by the GG kernel with parameter $\beta = 8$. Top: Original image and the ground truth blur kernel. Bottom: Highlighted deblurring results and their error maps computed w.r.t. the ground truth sharp images (1st - 4th columns) and the estimated blur kernels (5th column). Note that the kernels produced by individual algorithms have been resized for visualization.

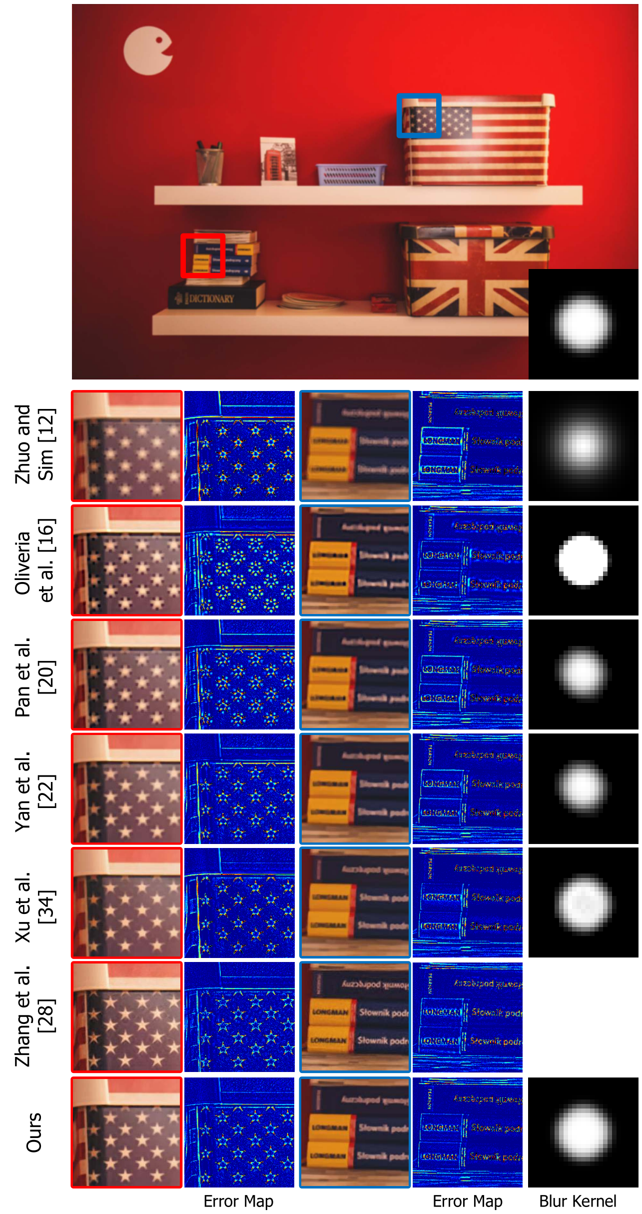

Fig. 21. The deblurring results on the synthetic images *Wall* blurred by the GG kernel with parameter $\beta = 6$. Top: Original image and the ground truth blur kernel. Bottom: Highlighted deblurring results and their error maps computed w.r.t. the ground truth sharp images (1st - 4th columns) and the estimated blur kernels (5th column). Note that the kernels produced by individual algorithms have been resized for visualization.

6×10^6 pixels. Fig. 22 shows that the blur kernels estimated by the two nonparametric algorithms, Pan *et al.* [20], Yan *et al.* [22], and Xu *et al.* [34] are far from a reasonable out-of-focus blur kernel. Accordingly, the produced deblurring images contain ringing artifacts. Oliveira *at al.* [16] fails in estimating the blur kernel because the required assumption on the image spectrum does not hold in this scene. As the out-of-focus blur kernel cannot be approximated by Gaussian function, the result produced by Zhuo and Sim [12] contains obvious ringing effect. The result by Zhang *et al.* [28] is not satisfactory due to the unpleasant noises. Our algorithm performs well in keeping sharp details and avoiding artifacts.

Fig. 23 shows the deblurring result of a blurry stone landmark. Fig. 24 shows the deblurred image of a natural scene with depth of field (DOF) change. It is observed that blur kernels estimated by Oliveira *at al.* [16], Pan *et al.* [20], Yan *et al.* [22], and Xu *et al.* [34] are incorrect, and their corresponding deblurred images are not satisfactory. As the out-of-focus blur cannot be approximated by Gaussian function, Zhuo and Sim [12] produces artifacts along edges. The deblurred image produced by Zhang *et al.* [28] contains high-frequency noises. In contrast, the deblurred image generated by our algorithm outperforms the competitors.

In the following, we analyze the reason why Zhang *et al.* [28] performs well on synthetic images but generates obvious high-frequency noises on real scene images. We evaluate the algorithm on images with the same scene but different image

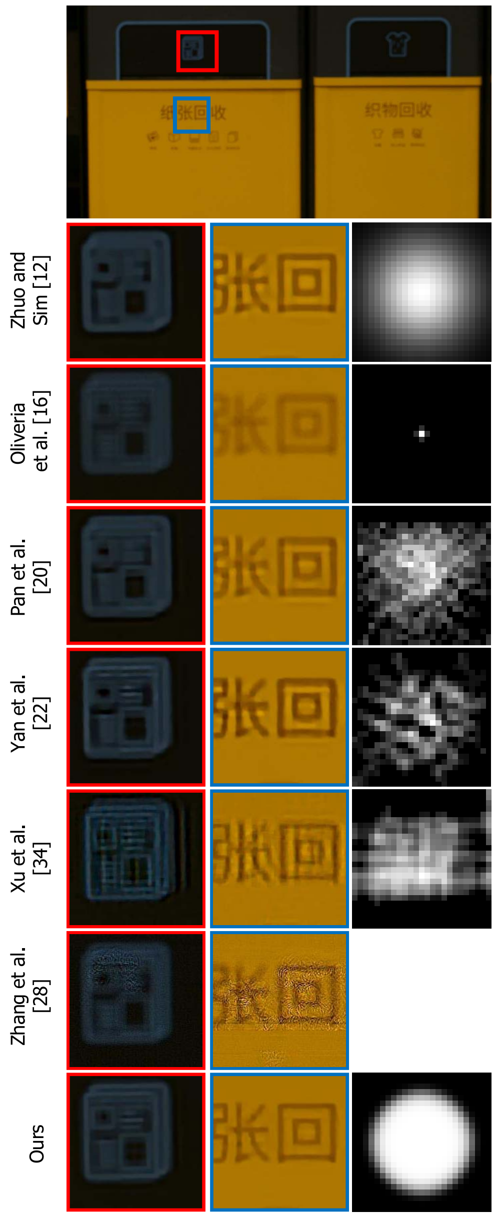

Fig. 22. The deblurring results on a real scene. Top: Blurred image. Bottom: Highlighted deblurring results (1st - 2nd columns) and the estimated blur kernels (3rd column). Note that the kernels produced by individual algorithms have been resized for visualization.

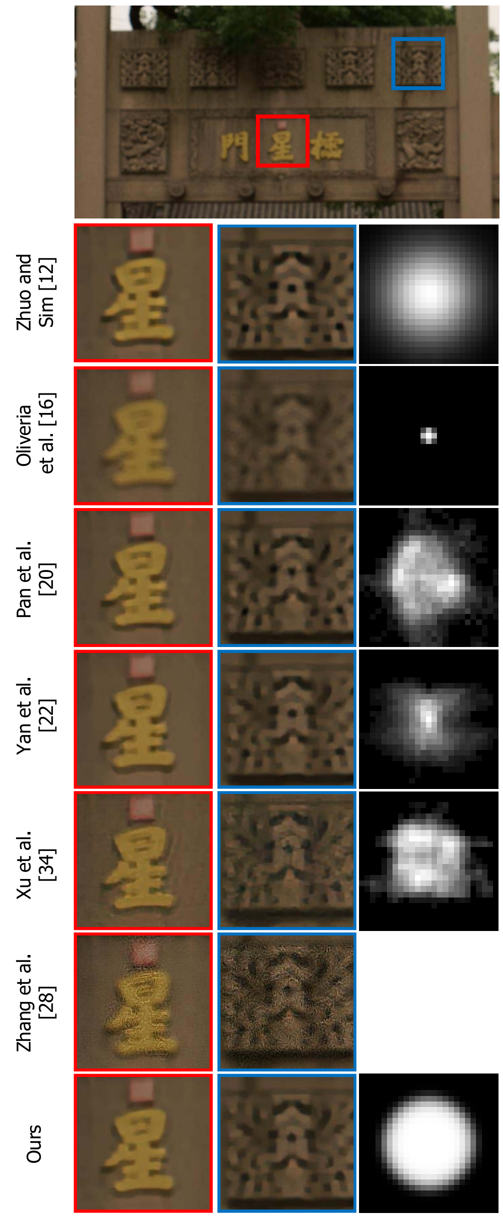

Fig. 23. The deblurring results on a real scene. Top: Blurred image. Bottom: Highlighted deblurring results (1st - 2nd columns) and the estimated blur kernels (3rd column). Note that the kernels produced by individual algorithms have been resized for visualization.

sizes (150×150 and 300×300 pixels). Fig. 25 clearly shows that the artificial noise is much severer on the larger image. The possible reason is that the debluring network in [28] is trained on small-sized images, and does not generalize well to large-sized real scene images used in the experiments. This is the limitation of learning-based image deblurring algorithms.

E. Computational Time

We also compare the computational time of kernel estimation in the deblurring algorithms except Zhang *et al.* [28] as it does not generate blur kernels. The computation is performed using MATLAB® on a Personal Computer with an Intel® i5 CPU and 16 GB memory. Table IV lists the computational times on an images with 1000×667 pixels. Pan *et al.* [20] and Yan *et al.* [22] are relatively slow, requiring 73.4 and 90.0 seconds, respectively. The reason is that these two algorithms compute the sharp image using alternative optimization and estimate the blur kernel in a coarse-to-fine manner. Xu *et al.* [34] requires 44.1 seconds, mostly spending on sharp edge restoration and alternative optimization. By estimating the kernel parameter in the frequency domain, Oliveria *et al.* [16] reduces the running time to 3.4 seconds. In comparison, our algorithm and Zhuo and Sim [12] are the fastest ones, thanks to the fact that locating edge patches and parameter estimation are both computationally efficient.

F. Extension to Nonuniform Deblurring

Although our algorithm computes one kernel parameter from an image, it can be easily extended to nonuniform

TABLE IV
COMPUTATIONAL TIME (IN SECONDS) OF KERNEL ESTIMATION IN DIFFERENT DEBLURRING ALGORITHMS

Oliveira *et al.* [16]	Zhuo and Sim [12]	Pan *et al.* [20]	Yan *et al.* [22]	Xu *et al.* [34]	Ours
3.4	0.5	73.4	90.0	44.1	0.5

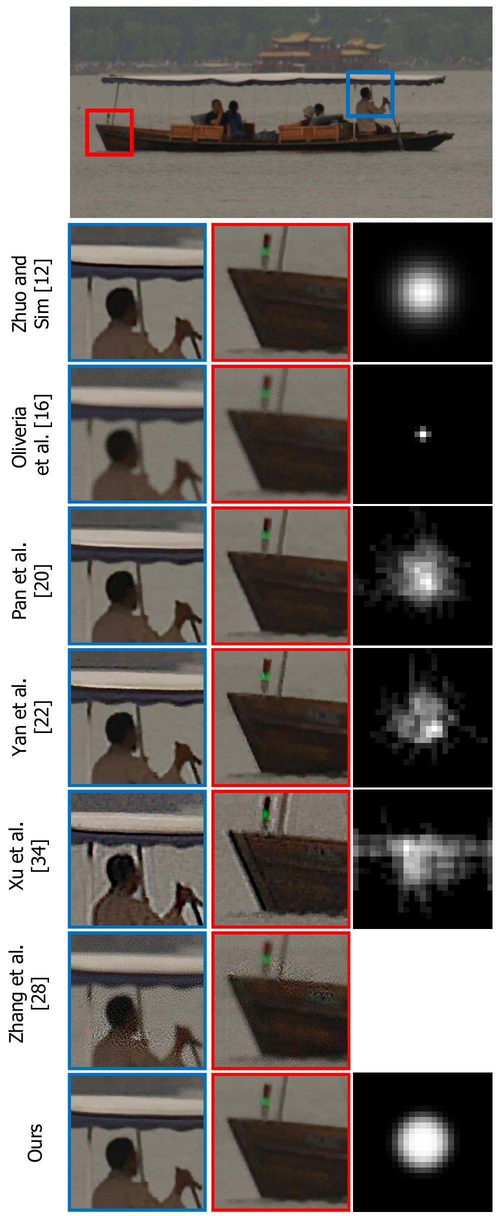

Fig. 24. The deblurring results on a real scene. Top: Blurred image. Bottom: Highlighted deblurring results (1st - 2nd columns) and the estimated blur kernels (3rd column). Note that the kernels produced by individual algorithms have been resized for visualization.

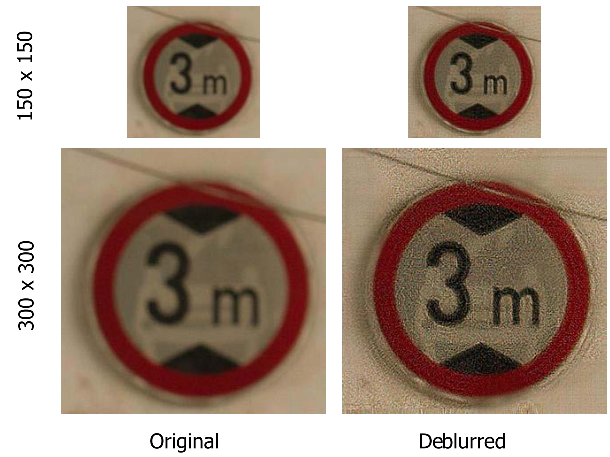

Fig. 25. Comparison of deblurring results produced by Zhang *et al.* [28] on images of sizes 150×150 and 300×300 pixels.

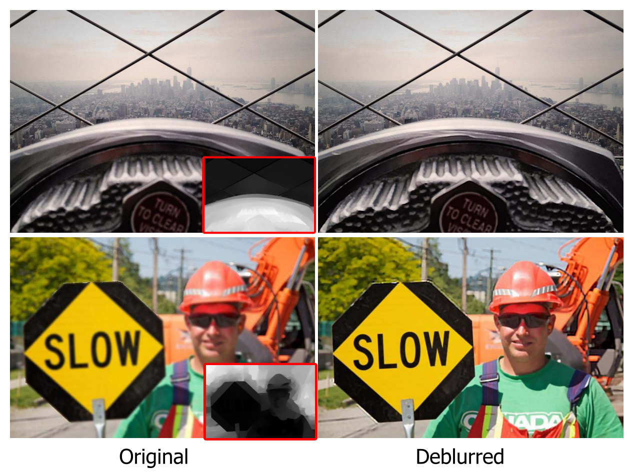

Fig. 26. Two nonuniform deblurring results using our algorithm. The insets show the defocus maps computed according to [49].

image deblurring after computing the defocus maps [49] of the images. Let $\mathbf{m}_i$ be the mask of the ith defocus level, and $\mathbf{h}_i$ the corresponding blur kernel, then the blurred image can be formulated as

$$\mathbf{f} = \sum_i \mathbf{h}_i * (\mathbf{m}_i \circ \mathbf{g}), \tag{21}$$

where the computation $\mathbf{m}_i \circ \mathbf{g}$ produces the image region at the ith defocus level. We compute the kernel parameter β_i from each defocus region, and deblur this region using our algorithm. By concatenating the regional deblurring results we finally obtain the complete sharp image. Two exemplar results of nonuniform image deblurring are illustrated in Fig. 26. It is observed that the image scenes at various distances all become sharper after applying our deblurring algorithm.

VI. CONCLUSION

In this paper, we have proposed an algorithm of estimating generalized Gaussian (GG) blur kernels for out-of-focus image deblurring. The GG model is simplified to its single-parameter form thanks to the shape characteristics of real out-of-focus blur kernels. The kernel parameter is computed using a gradient-based approach with proper edge patch selection. It has been validated by experiments that the GG functions are more appropriate in depicting out-of-focus blur kernels, and outperforms both parametric or nonparametric kernel estimation approaches.

A limitation of our algorithm is that the blur kernel is only estimated on an image or a region rather than a small patch. We plan to extend our kernel estimation algorithm for dense nonuniform image deblurring, with emphasis on estimating the GG parameter from more general edges using deep learning.

REFERENCES

[1] Y. Wang *et al.*, "AllFocus: Patch-based video Out-of-Focus blur reconstruction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 9, pp. 1895–1908, Sep. 2017.

[2] S. Lefkimmiatis, A. Bourquard, and M. Unser, "Hessian-based norm regularization for image restoration with biomedical applications," *IEEE Trans. Image Process.*, vol. 21, no. 3, pp. 983–995, Mar. 2012.

[3] F. Hua, P. Johnson, N. Sazonova, P. Lopez-Meyer, and S. Schuckers, "Impact of out-of-focus blur on face recognition performance based on modular transfer function," in *Proc. 5th IAPR Int. Conf. Biometrics (ICB)*, Mar. 2012, pp. 85–90.

[4] P. Campisi and K. Egiazarian, *Blind Image Deconvolution: Theory and Applications*. Boca Raton, FL, USA: CRC Press, 2016.

[5] S. Zhuo, D. Guo, and T. Sim, "Robust flash deblurring," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2440–2447.

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

[7] 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.

[8] D. Krishnan, T. Tay, and R. Fergus, "Blind deconvolution using a normalized sparsity measure," in *Proc. CVPR*, Jun. 2011, pp. 233–240.

[9] L. Chen, F. Fang, T. Wang, and G. Zhang, "Blind image deblurring with local maximum gradient prior," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1742–1750.

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

[11] A. E. Savakis and H. J. Trussell, "On the accuracy of PSF representation in image restoration," *IEEE Trans. Image Process.*, vol. 2, no. 2, pp. 252–259, Apr. 1993.

[12] S. Zhuo and T. Sim, "Defocus map estimation from a single image," *Pattern Recognit.*, vol. 44, no. 9, pp. 1852–1858, Sep. 2011.

[13] M. Gokstorp, "Computing depth from out-of-focus blur using a local frequency representation," in *Proc. 12th Int. Conf. Pattern Recognit.*, vol. 1, 1994, pp. 153–158.

[14] Y.-W. Tai and M. S. Brown, "Single image defocus map estimation using local contrast prior," in *Proc. 16th IEEE Int. Conf. Image Process. (ICIP)*, Nov. 2009, pp. 1797–1800.

[15] M. E. Moghaddam, "A mathematical model to estimate out of focus blur," in *Proc. 5th Int. Symp. Image Signal Process. Anal.*, Sep. 2007, pp. 278–281.

[16] J. P. Oliveira, M. A. T. Figueiredo, and J. M. Bioucas-Dias, "Parametric blur estimation for blind restoration of natural images: Linear motion and Out-of-Focus," *IEEE Trans. Image Process.*, vol. 23, no. 1, pp. 466–477, Jan. 2014.

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

[18] 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, Jul. 2006.

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

[20] 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.

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

[22] 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. (CVPR)*, Jul. 2017, pp. 4003–4011.

[23] Q. Shan, J. Jia, and A. Agarwala, "High-quality motion deblurring from a single image," *ACM Trans. Graph.*, vol. 27, no. 3, pp. 1–10, Aug. 2008.

[24] Y. Bai, H. Jia, M. Jiang, X. Liu, X. Xie, and W. Gao, "Single image blind deblurring using multi-scale latent structure prior," *IEEE Trans. Circuits Syst. Video Technol.*, May 27, 2019, doi: 10.1109/TCSVT.2019.2919159.

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

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

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

[28] J. Zhang *et al.*, "Dynamic scene deblurring using spatially variant recurrent neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2521–2529.

[29] J. Shi, L. Xu, and J. Jia, "Just noticeable defocus blur detection and estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 657–665.

[30] S.-J. Chen and H.-L. Shen, "Multispectral image Out-of-Focus deblurring using interchannel correlation," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 4433–4445, Nov. 2015.

[31] T.-Y. Sun, S.-J. Ciou, C.-C. Liu, and C.-L. Huo, "Out-of-focus blur estimation for blind image deconvolution: Using particle swarm optimization," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, Oct. 2009, pp. 1627–1632.

[32] S. Metari and F. Deschenes, "A new convolution kernel for atmospheric point spread function applied to computer vision," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, Oct. 2007, pp. 1–8.

[33] W.-S. Lai, J.-B. Huang, Z. Hu, N. Ahuja, and M.-H. Yang, "A comparative study for single image blind deblurring," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1701–1709.

[34] X. Xu, J. Pan, Y.-J. Zhang, and M.-H. Yang, "Motion blur kernel estimation via deep learning," *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 194–205, Jan. 2018.

[35] S. Su, M. Delbracio, J. Wang, G. Sapiro, W. Heidrich, and O. Wang, "Deep video deblurring for hand-held cameras," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1279–1288.

[36] K. Zhang, W. Luo, Y. Zhong, L. Ma, W. Liu, and H. Li, "Adversarial spatio-temporal learning for video deblurring," *IEEE Trans. Image Process.*, vol. 28, no. 1, pp. 291–301, Jan. 2019.

[37] T. Darrell and K. Wohn, "Pyramid based depth from focus," in *Proc. Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 1988, pp. 504–509.

[38] J. H. Elder and S. W. Zucker, "Local scale control for edge detection and blur estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 7, pp. 699–716, Jul. 1998.

[39] W. J. Smith, *Modern Optical Engineering*. New York, NY, USA: McGraw-Hill, 2008.

[40] E. Hecht, *Optics*. London, U.K.: Pearson, 2017.

[41] J. Kotera, F. Šroubek, and P. Milanfar, "Blind deconvolution using alternating maximum a posteriori estimation with heavy-tailed priors," in *Proc. Int. Conf. Comput. Anal. Images Patterns*. Berlin, Germany: Springer, 2013, pp. 59–66.

[42] S.-J. Chen, H.-L. Shen, C. Li, and J. H. Xin, "Normalized total gradient: A new measure for multispectral image registration," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1297–1310, Mar. 2018.

[43] J. Canny, "A computational approach to edge detection," in *Readings in Computer Vision*. Amsterdam, The Netherlands: Elsevier, 1987, pp. 184–203.

[44] J.-C. Yoo and T. H. Han, "Fast normalized cross-correlation," *Circuits, Syst. Signal Process.*, vol. 28, no. 6, p. 819, 2009.

[45] R. Franzen. (Apr. 1999). *Kodak Lossless True Color Image Suite*. [Online]. Available: http://r0k. us/graphics/kodak

[46] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011, doi: 10.1109/TPAMI.2010.161.

[47] T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2014, pp. 740–755.
[48] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[49] J. Park, Y.-W. Tai, D. Cho, and I. S. Kweon, "A unified approach of multi-scale deep and hand-crafted features for defocus estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1736–1745.

Yu-Qi Liu received the B.Eng. degree in automation science and technology from Xi'an Jiaotong University, China, and the M.Sc. degree in communications and signal processing from Imperial College London, U.K., in 2015 and 2016, respectively. He is currently pursuing the Ph.D. degree with the College of Information Science and Electronic Engineering, Zhejiang University, China. His research interests are image deblurring and image processing.

Xin Du received the B.Eng. degree in electronic engineering from Hangzhou University, China, in 1997, and the M.Eng. and Ph.D. degrees from Zhejiang University, China, in 2000 and 2003, respectively. He is currently an Associate Professor with the College of Information Science and Electronic Engineering, Zhejiang University. His research interests are image/video processing and computer vision.

Hui-Liang Shen (Member, IEEE) received the B.Eng. and Ph.D. degrees in electronic engineering from Zhejiang University, China, in 1996 and 2002, respectively. He was a Research Associate and a Research Fellow with The Hong Kong Polytechnic University from 2001 to 2005. He is currently a Professor with the College of Information Science and Electronic Engineering, Zhejiang University. His research interests include multispectral imaging, image processing, and computer vision.

Shu-Jie Chen received the B.Eng. and Ph.D. degrees in electronic engineering from Zhejiang University, China, in 2012 and 2018, respectively. He is currently an Assistant Professor with the School of Computer and Information Engineering, Zhejiang Gongshang University, Hangzhou, China. His research interests are multispectral imaging and image processing.