Robust Estimation of Motion Blur Kernel Using a Piecewise-Linear Model

Sungchan Oh, *Student Member, IEEE*, and Gyeonghwan Kim, *Member, IEEE*

***Abstract*— Blur kernel estimation is a crucial step in the deblurring process for images. Estimation of the kernel, especially in the presence of noise, is easily perturbed, and the quality of the resulting deblurred images is hence degraded. Since every motion blur in a single exposure image can be represented by 2D parametric curves, we adopt a piecewise-linear model to approximate the curves for the reliable blur kernel estimation. The model is found to be an effective tradeoff between flexibility and robustness as it takes advantage of two extremes: 1) the generic model, represented by a discrete 2D function, which has a high degree of freedom (DOF) for the maximum flexibility but suffers from noise and 2) the linear model, which enhances robustness and simplicity but has limited expressiveness due to its low DOF. We evaluate several deblurring methods based on not only the generic model, but also the piecewise-linear model as an alternative. After analyzing the experiment results using real-world images with significant levels of noise and a benchmark data set, we conclude that the proposed model is not only robust with respect to noise, but also flexible in dealing with various types of blur.**

***Index Terms*— Deblur, blur kernel estimation, piecewise-linear curve, image reconstruction, snake, active contour.**

I. Introduction

ACQUIRING bright and clear images under dim lighting conditions is a challenging task. The brightness of the images can be determined by the sensor sensitivity (ISO), exposure time, and aperture. A higher ISO value, longer exposure time, and wider aperture setting increase the image brightness at the cost of image quality. A longer exposure time effectively ensures brightness of acquired image but leads to a higher probability of movement in both the scene and the camera while the shutter is open. Conversely, a higher ISO setting to obtain brighter image introduces more noise which is composed of time-independent and time-dependent components. The situation requires a higher ISO value implies a weak image signal and a relatively high level of time-independent noise which is inherent in the sensor. A shorter exposure time for compensating the high ISO results in time-dependent noise due to increased randomness of captured photons [1]. Quite often, since the incoming light is insufficient in dark situations, both noise and blur are present in the image.

The blur, which is assumed to be uniform over a discrete spatial domain, $\Omega_d \subset \mathbb{Z}^2$, can be expressed by the convolution operator, $\otimes$,

$$b(\mathbf{x}) = f(\mathbf{x}) \otimes h(\mathbf{x}) + n(\mathbf{x}), \ \mathbf{x} \in \Omega_d, \tag{1}$$

where $b(\mathbf{x})$, $f(\mathbf{x})$, $h(\mathbf{x})$, and $n(\mathbf{x})$ designate the blurred image, underlying latent image, blur kernel, and additive noise, respectively. The deblurring problem is highly ill-posed since it is composed of two unknown dependent variables, $f(\mathbf{x})$ and $h(\mathbf{x})$. Studies conducted on the deblurring of noisy images with known kernels [2], [3] show that the estimation of the kernel is a prerequisite for robust deblurring. As the deblurring process is usually conducted by *de-convolving* $b(\mathbf{x})$ with respect to $h(\mathbf{x})$, the estimation accuracy of $h(\mathbf{x})$ has a direct impact on the quality of the deblurred images. In addition, the randomness of the noise $n(\mathbf{x})$ is a major hurdle to overcome in the deblurring process, especially for the kernel estimation stage.

The diagram in Fig. 1 shows several models of the blur kernel $h(\mathbf{x})$ based on the degree of flexibility in representing the sensor translation of a moving camera, as illustrated in Fig. 2. In the generic model, every possible translation is discretized to $\mathbf{x} \in \Omega_d$,

$$h(\mathbf{x}) : \Omega_d \rightarrow \mathbb{R}. \tag{2}$$

When the amount of the translation is known to be no larger than $W \times W$, the representation in (2) becomes a W^2 DOF model. Due to its flexibility, the generic model is the most commonly used among the various deblurring methods [4]–[10]. However, estimation of the W^2 parameters is easily misled by the noise and clutter in $b(\mathbf{x})$. The bottom left box in Fig. 1 displays the results of the kernel estimation and deconvolution by Cho and Lee [8] for a given blurred image (top left box) in the presence of noise. As shown in the figure, the kernel estimation and deconvolution process is significantly affected by noise. The other extreme with the least flexibility is the linear model [11], [12] which has 2 DOF, one for the direction and one for the length of the sensor motion trajectory. As clearly demonstrated by the bottom right box in Fig. 1, the linear model is limited in its ability to represent fairly complex blurs in real-world images.

The limitation that the generic and the linear models suffer from can be regarded as a lack of trade-off between the flexibility and the robustness in the kernel representation. Therefore, we devise a model to relieve the limitation.

Manuscript received January 13, 2013; revised July 3, 2013, November 6, 2013, and January 9, 2014; accepted January 21, 2014. Date of publication January 29, 2014; date of current version February 18, 2014. This work was supported in part by the Samsung Electronics and in part by the Sogang Research Frontiers. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Farhan A. Baqai.

The authors are with the Department of Electronic Engineering, Sogang University, Seoul 100-611, Korea (e-mail: amuckrunner@hotmail.com; gkim@sogang.ac.kr).

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

Digital Object Identifier 10.1109/TIP.2014.2303637

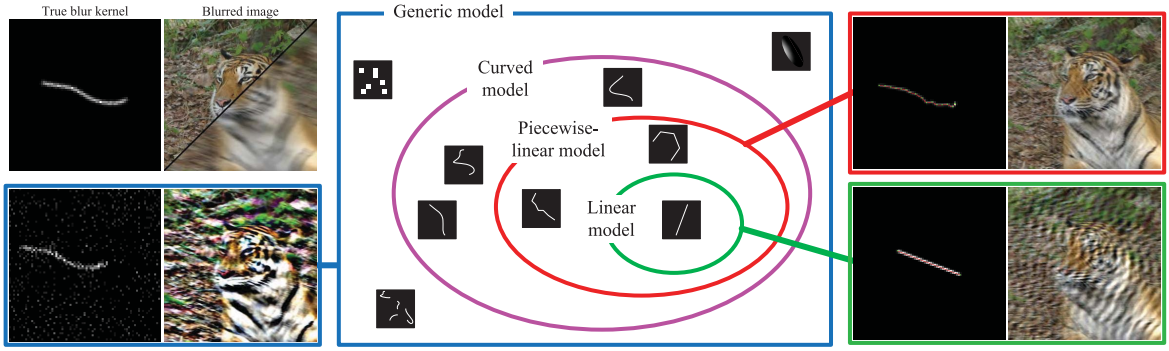


Fig. 1. Blur kernel models and their deblurring results by [8]. The top left box shows the input blurred image contaminated by Gaussian noise ($\sigma = 2.5$) and the corresponding blur kernel. The generic blur kernel model, which has a 75×75 DOF, and the corresponding deblurred image are found in the bottom left box. The bottom right box shows the results of the 2-DOF linear model. The piecewise-linear model in the top right box has 34 DOF (17 control points), which is in between the DOF's of the generic and the linear models.

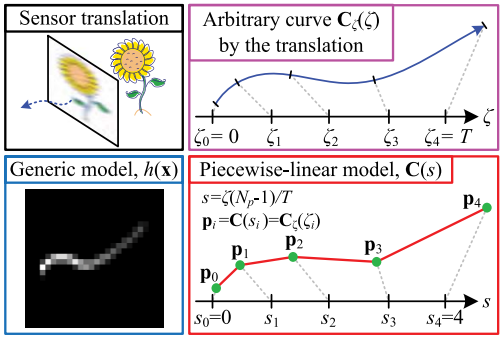


Fig. 2. Sensor translation by camera shake and models for blur kernel representation. $N_p = 5$ in this illustration.

As shown in the top right box of Fig. 2, the translation of the sensor plane during exposure time T can be expressed as an arbitrary parametric curve in a 2-dimensional continuous domain $\mathbf{\Omega}_c \subset \mathbb{R}^2$,

$$\mathbf{C}_\zeta(\zeta) : [0, T] \rightarrow \mathbf{\Omega}_c.$$

The parametric curved model is able to express the motion trajectory of the sensor without being overly flexible as is the limitation with the generic model. To avoid the numerical intractability due to an infinite DOF of $\mathbf{C}_\zeta(\zeta)$, it is then further approximated using a piecewise-linear curve by connecting the control points $\mathbf{p}_i = [p_i^{(x)}, p_i^{(y)}]^T = \mathbf{C}_\zeta(\zeta_i), i = 0, \ldots, N_p - 1$, where ζ_i are N_p equidistant samples subdviding $[0, T]$, as shown in the bottom right box of Fig. 2. Since the proposed model is fully determined by the N_p control points, it has $2N_p$ DOF.

When N_p is set to be sufficiently large, the piecewise-linear curve approximates $\mathbf{C}_\zeta(\zeta)$ within acceptable limits, especially for discrete blur kernels which are relatively small compared to $b(\mathbf{x})$. Furthermore, the DOF of $2N_p$ can be much smaller than W^2 of the generic model, to achieve an effective compromise between the flexibility of the generic model and the robustness of the linear model, as observed in the top right box in Fig. 1. In this paper, we present a complete framework for the representation and manipulation of the piecewise-linear blur kernel, as well as for the estimation of the curve with physically-sound regularization schemes.

A. Prior Work

There have been several researches which have employed the generic model shown in (2). Fergus *et al.* [4] and Shan *et al.* [5] proposed blind deblurring algorithms based on probabilistic inferences. These algorithms take a single blurred image as the observation. In the fast motion deblurring method proposed by Cho and Lee [8], iterative sharpening by shock filtering is employed and spurious image gradients are utilized to predict the latent image. By the prediction step, Cho and Lee [8] were able to significantly reduce the computational burden. There have also been efforts to mitigate the difficulties in deblurring by modifying the image acquisition stage. For example, in Zhuo *et al.* [7], a flash image is acquired along with the blurred image. The sharper edges in the flash image are incorporated into the deblurring process while suppressing distortions in the image brightness. Similarly, Yuan *et al.* [6] used a noisy image to guide kernel estimation. In Tico and Vehvilainen [13], multiple frames of the blurred images were used for more effective image deblurring.

Regarding the linear model-based approaches, Yitzhaky and Kopeika [14] blindly estimated the linear blurs by identifying the directional correlation function of the blurred images. Ji and Liu [15] also introduced the idea of defining the kernel as a straight line segment with a varying intensity profile. The kernel is estimated by finding sharp peaks in the cepstrum domain to determine the orientation and the length of the line segment, and then modifying the intensity profile along the segment for a better fit with the true blur kernel. Rav-Acha and Peleg [11] employed a linear model to estimate two straight kernels from a pair of differently blurred images. However, the overly simplified representation makes the linear models unsuitable for general motion blurs, as seen in the bottom right box of Fig. 1.

In the approaches based on the piecewise-linear models, proposed by Patanukhom and Nishihara [16] and Kang *et al.* [17], distinct motion components are found independently based on the spectral analysis of an image. The components are then assembled to form a complete kernel. Since the methods rely heavily on the distinctiveness of the motion directions, they are limited in their ability to estimate smoothly curved blur kernels. Moreover, it is difficult to be adopted

to general deblurring methods as their kernel representations are closely related to their blind blur identification schemes. Ben-Ezra and Nayar [18] employed a piecewise-linear model for deblurring using a special hybrid camera. In their method, the motion components were estimated based on secondarily acquired image sequences.

Recently, non-uniform deblurring methods which extend the translational motion blurs in 2-dimensional spaces into 3-dimensional ones with additional in-plane rotations have been proposed. In Whyte *et al.* [19], 3-dimensional space is used in representing in-plane and out-of-plane camera rotations, and Gupta *et al.* [20] and Hirsch *et al.* [21] approximate the blurs using 2-dimensional translation and additional in-plane rotation. In spite of its flexibility, the method of Whyte *et al.* [19] shows weakness in estimating large blurs in Köhler *et al.* [22]. The weakness stems from the fact that the blur is treated as a generic model in 3-dimensional space with a significantly increased DOF.

B. Problem Formulation

The piecewise-linear curve is defined over a continuous parameter domain $\left[0, N_p - 1\right]$, i.e. $\mathbf{C}(s) : \left[0, N_p - 1\right] \rightarrow \Omega_c$, where $s \in \left[0, N_p - 1\right]$ is linearly mapped from $\zeta \in [0, T]$, such that $s = \zeta \left(N_p - 1\right)/T$. Thus, $\mathbf{C}(s)$ is equivalent to a concatenation of parametric line segments $\mathbf{l}(\cdot)$, connecting $\mathbf{p}_i$

$$\mathbf{C}(s; \mathbf{Q}) = \mathbf{l}(s - s_i; \mathbf{p}_i, \mathbf{p}_{i+1}), \; s_i \leq s \leq s_{i+1}, \tag{3}$$

where $\mathbf{l}(\rho; \mathbf{x}_1, \mathbf{x}_2) = (1 - \rho)\mathbf{x}_1 + \rho\mathbf{x}_2, \rho \in [0, 1]$ and $s_i = \zeta_i \left(N_p - 1\right)/T$. In (3), the set of N_p control points is abbreviated by the $2N_p$ vector $\mathbf{Q} = [\ldots, p_i^{(x)}, p_i^{(y)}, \ldots]^T$.

The piecewise-linear curve as defined in (3), which is a set of coordinates in Ω_c continuously mapped from $\left[0, N_p - 1\right] \subset \mathbb{R}$, cannot directly replace $h(\mathbf{x})$ in (1). Therefore, a proper conversion of the piecewise-linear curve $\mathbf{C}(s; \mathbf{Q})$ into a form,

$$k(\mathbf{x}; \mathbf{Q}) : \Omega_d \rightarrow \mathbb{R}, \tag{4}$$

is necessary. Section II is on the conversion in greater detail.

Adopting the representation in (4) and setting $\mathbf{Q}$ as unknown parameters to be estimated, general deblurring methods can be described by alternating optimization between kernel estimation and deconvolution, as summarized in Algorithm 1.

The energy terms $J_K(\hat{f}, \mathbf{Q})$ and $J_I(f, \hat{\mathbf{Q}})$ in Algorithm 1 are defined as

$$J_K\left(\hat{f}, \mathbf{Q}\right) = J_D\left(\hat{f}, \mathbf{Q}\right) + J_R(\mathbf{Q}), \tag{5}$$

$$J_I\left(f, \hat{\mathbf{Q}}\right) = J_D\left(f, \hat{\mathbf{Q}}\right) + J_F(f), \tag{6}$$

where $J_R(\mathbf{Q})$ and $J_F(f)$ are regularization energy terms for $\mathbf{Q}$ and f, respectively. The data term J_D contained in (5) and (6) are usually defined quadratically,

$$J_D(f, \mathbf{Q}) = \frac{1}{2} \sum\nolimits_{\mathbf{x} \in \Omega_d} \|r(\mathbf{x}; \mathbf{Q})\|^2, \tag{7}$$

where $\mathrm{r}(\mathbf{x}; \mathbf{Q}) = f(\mathbf{x}) \otimes k(\mathbf{x}; \mathbf{Q}) - b(\mathbf{x})$. While the data energy J_D in (7) reflects the fitness of $f(\mathbf{x})$ and $k(\mathbf{x}; \mathbf{Q})$ to $b(\mathbf{x})$, the regularization energy terms are added to make the problem of estimating $k(\mathbf{x}; \mathbf{Q})$ and $f(\mathbf{x})$ well-posed. Sparsity has been widely utilized to specify the regularization. The sparsity of the image gradients is used in the regularization energy,

$$J_F(f(\mathbf{x})) = \lambda_F \sum\nolimits_{\mathbf{x} \in \Omega_d} \|\nabla f(\mathbf{x})\|^{\gamma_F}, \tag{8}$$

where λ_F is a weighting parameter and the exponent γ_F reflects the desired distribution of $\nabla f(\mathbf{x})$, i.e. $\gamma_F = 2$, $\gamma_F = 1$, and $0 < \gamma_F < 1$ for the Gaussian, the Laplacian, and the hyper-Laplacian distributions, respectively [23]. In the estimation of the generic model, the sparsity is also adopted to suppress spurious entries of $h(\mathbf{x})$. Whereas the proposed model in (4) is controlled solely by $\mathbf{p}_i$, in contrast to the generic model in (2), the regularization energy $J_R(\mathbf{Q})$ should be defined differently.

The proposed regularization energy $J_R(\mathbf{Q})$, specialized to the piecewise-linear blur kernel, is presented in Section III. The effective minimization scheme for the energy in (5) is also described in Section III. Section IV is dedicated to experiments in which the proposed model is evaluated by replacing the kernels of several deblurring methods with ones obtained by the model. In order to improve computational efficiency, a method of directly manipulating $\mathcal{DFT}\{k(\mathbf{x}; \mathbf{Q})\}$, discrete Fourier transform (DFT) of $k(\mathbf{x}; \mathbf{Q})$, without explicit application of DFT is presented in the Appendix.

Algorithm 1 $SingleLevelDeblur(f_{init}, \mathbf{Q}_{init}, b)$

1: $\hat{\mathbf{Q}} = \mathbf{Q}_{init}$, $\hat{f} = predict(f_{init})$
2: **for** $iter = 1$ to N_{iter} **do**
3: Estimate kernel, $\hat{\mathbf{Q}} = \mathrm{argmin}_{\mathbf{Q}}\{J_K(\hat{f}, \mathbf{Q})\}$
4: Deconvolution, $\hat{f} = \mathrm{argmin}_f\{J_I(f, \hat{\mathbf{Q}})\}$
5: $\hat{f} = predict(\hat{f})$
6: **end for**
7: **return** $\hat{f}$ and $\hat{\mathbf{Q}}$

in line 3: In Yuan *et al.* [7], $\hat{f}$ is replaced by the noisy image and $N_{iter} = 1$.
in line 4: The noisy image and the flash image are used as guidance of deconvolution in Yuan *et al.* [6] and Zhuo *et al.* [7], respectively.
in line 1 and 5: $predict(f) = f$ in [6], [7]. In Cho and Lee [8], $predict()$ is implemented by the bilateral filter, shock filter and edge pruning.

II. Piecewise-Linear Representation of Blur Kernel

The sequence of mathematical operations toward $k(\mathbf{x}; \mathbf{Q})$ in (4) is summarized in (9).

$$\begin{aligned} k(\mathbf{x}; \mathbf{Q}) &= \mathcal{D}\{\mathcal{P}\{\mathbf{C}(s; \mathbf{Q})\}\}, \\ \text{where } \; \mathcal{P}\{\mathbf{C}(s; \mathbf{Q})\} &: \mathbf{C}(s; \mathbf{Q}) \mapsto \kappa(\boldsymbol{\chi}; \mathbf{Q}), \\ \mathcal{D}\{\kappa(\boldsymbol{\chi}; \mathbf{Q})\} &: \kappa(\boldsymbol{\chi}; \mathbf{Q}) \mapsto k(\mathbf{x}; \mathbf{Q}) \end{aligned} \tag{9}$$

In (9), $\boldsymbol{\chi} = [\chi, \xi]^T \in \Omega_c$ and $\mathbf{x} = [x, y]^T \in \Omega_d$ are continuous and discrete spatial domain coordinates, respectively. The projection operator $\mathcal{P}\{\cdot\}$ maps a curve $\mathbf{C}(s; \mathbf{Q})$ to a continuous kernel $\kappa(\boldsymbol{\chi}; \mathbf{Q}) : \Omega_c \rightarrow \mathbb{R}$ and is sampled by the discretization operator $\mathcal{D}\{\cdot\}$ to obtain the discrete kernel, $k(\mathbf{x}; \mathbf{Q}) : \Omega_d \rightarrow \mathbb{R}$.

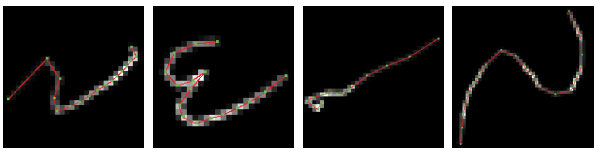


Fig. 3. Various configurations of the control points and the corresponding blur kernels. From left to right, N_p is set to be 17, 17, 17, and 33. In accordance with the rest of this paper, the green dots, red lines, and gray squares designate the control points, line segments connecting the points, and entries in $k(\mathbf{x};\mathbf{Q})$, respectively.

As defined in (3), $\mathbf{C}(s;\mathbf{Q})$ consists of (N_p-1) line segments. Thus, $\kappa(\boldsymbol{\chi};\mathbf{Q})$ can be divided into parts, such that

$$\kappa(\boldsymbol{\chi};\mathbf{Q}) = \frac{1}{N_p-1}\sum_{i=0}^{N_p-2}\kappa_i(\boldsymbol{\chi};\mathbf{Q}), \tag{10}$$

where $1/(N_p-1)$ is a normalizing constant and $\kappa_i(\boldsymbol{\chi};\mathbf{Q})$ is a fraction of $\kappa(\boldsymbol{\chi};\mathbf{Q})$ produced by the i-th line segment, $\mathbf{l}(\rho;\mathbf{p}_i,\mathbf{p}_{i+1})$. As we regard $\kappa(\boldsymbol{\chi};\mathbf{Q})$ in (4) and (10) as a collection of impulses placed along $\mathbf{C}(s:\mathbf{Q})$, $\kappa_i(\boldsymbol{\chi};\mathbf{Q})$ is obtained by defining $\mathcal{P}\{\cdot\}$ as an integral operator such that,

$$\begin{aligned}\kappa_i(\boldsymbol{\chi};\mathbf{Q}) &= \mathcal{P}\{\mathbf{l}(\rho;\mathbf{p}_i,\mathbf{p}_{i+1})\}\\ &= \int_0^1 \delta(\boldsymbol{\chi}-\mathbf{l}(\rho;\mathbf{p}_i,\mathbf{p}_{i+1}))\,\mathrm{d}\rho,\end{aligned}$$

where $\delta(\boldsymbol{\chi}-\mathbf{l})$ denotes the Dirac impulse located at $\mathbf{l}$. Note that $\kappa_i(\boldsymbol{\chi};\mathbf{Q})$ has a constant profile along $\mathbf{l}(\rho)$, i.e. $\kappa_i(\mathbf{l}(\rho);\mathbf{Q}) = 1/\|\mathbf{p}_{i+1}-\mathbf{p}_i\|$, $\forall\rho\in[0,1]$, and $\int\kappa_i(\boldsymbol{\chi};\mathbf{Q})\,\mathrm{d}\boldsymbol{\chi} = 1$, regardless of $\mathbf{Q}$. As a consequence, $\int\kappa(\boldsymbol{\chi};\mathbf{Q})\,\mathrm{d}\boldsymbol{\chi} = 1$ due to the normalization in (10). This aspect will be further discussed later in this section.

In general, the impulses in $\kappa(\boldsymbol{\chi};\mathbf{Q})$ have sub-pixel locations, and they are not captured by the grid of sampling impulses. Therefore, a kernel $\Lambda(\boldsymbol{\chi})$ is convolved before the sampling. $k(\mathbf{x};\mathbf{Q})$ is then defined by $\mathcal{D}\{\cdot\}$,

$$\begin{aligned}k(\mathbf{x};\mathbf{Q}) &= \mathcal{D}\{\kappa(\boldsymbol{\chi};\mathbf{Q})\}\\ &= \int_{\Omega_c}\delta(\boldsymbol{\chi}-\mathbf{x})\cdot(\Lambda(\boldsymbol{\chi})\otimes\kappa(\boldsymbol{\chi};\mathbf{Q}))\,\mathrm{d}\boldsymbol{\chi}.\end{aligned} \tag{11}$$

In (11), the spacing of the sampling grid is assumed to be one, for simplicity. Possible choices for $\Lambda(\boldsymbol{\chi})$ can be either rectangular or triangular kernels, which correspond to the zero-order hold and the first-order hold, respectively. As two separate points in sub-grid proximity cannot be distinguished by the zero-order hold, the first-order hold sampling is employed by setting $\Lambda(\boldsymbol{\chi})$ to be a bilinear kernel, as in (12).

$$\begin{aligned}\Lambda(\chi,\xi) &= \mathrm{tri}(\chi)\otimes\mathrm{tri}(\xi),\\ \text{where } \mathrm{tri}(\chi) &= \begin{cases}1-|\chi|, & |\chi|\le 1\\ 0, & \text{otherwise}\end{cases}.\end{aligned} \tag{12}$$

Fig. 3 shows examples of kernels generated using the projection and the discretization operators, with various configurations of control points. Although the kernels are generated from a small number of control points, they are expressive enough to cover a diverse range of motion blurs in the real world. Note that contribution of each line segment to $k(\mathbf{x};\mathbf{Q})$ is $1/(N_p-1)$, out of 1 in total. Thus, a shorter line segment induces a higher values of $k(\mathbf{x};\mathbf{Q})$ at the corresponding location, and vice versa. This effectively reflects the varying speeds of the camera's motion as shown in Fig. 3.

There are some properties that need to be pursued by a generic model based kernel estimation. They are *non-negativity* ($h(\mathbf{x})\ge 0$, $\forall\mathbf{x}$) which is motivated by the non-negative light integration during the exposure, *energy preservation* ($\sum_{\mathbf{x}\in\Omega_d}h(\mathbf{x})=1$) for maintaining the brightness of the blurred image in the deconvolution, and the *sparsity* prior on the kernels. The generic model [4]–[8] does not satisfy non-negativity and energy preservation by itself. Thus, the renormalization process, which consists of rejecting negative entries and normalization, is used in estimation of $h(\mathbf{x})$. The sparsity prior is realized using an additional penalty term, which is similar to (8),

$$J_H(h(\mathbf{x})) = \lambda_H\sum_{\mathbf{x}\in\Omega_d}\|h(\mathbf{x})\|^{\gamma_H}. \tag{13}$$

In contrast to the generic model, the proposed model in (9), which is generated by positive integration of sharp impulses and normalization, inherently satisfies the necessary properties. Moreover, the proposed model is solely defined by the configuration of $\mathbf{p}_i$, it is irrelevant to (13).

III. Blur Kernel Estimation

In spite of the prescribed properties of the proposed model, estimation of the piecewise-linear blur kernel is an ill-posed problem not only for the ill-posedness of general kernel estimation problem itself, but also because of non-uniqueness of the points' configuration w.r.t. a fixed kernel. In that regards, based on the physical restraints of blurring, the regularization energy $J_R(\mathbf{Q})$ in (5) is proposed in this section. Efficient energy minimization methods for the kernel estimation are also provided.

A. Regularization of Piecewise-Linear Blur Kernel

Since physical cameras have non-zero masses, rather than being able to move freely, they are limited in their motions due to inertia. Thus a camera's motion trajectory draws a smooth curve in terms of *parametrization* and *curvature* which reflect variations in the speed and direction of the motion, respectively [24]. Based on the physical constraints, we propose a curve regularization energy $J_R(\mathbf{Q})$ by employing the idea suggested by Delingette and Montagnat [25].

Kass *et al.* [26] developed a method called *Snakes* which estimates a curve representing an object boundary. In Kass *et al.* [26], tension and stiffness energies account for regularizing parametrization and curvature of the curve, respectively. The curve regularization method in Delingette and Montagnat [25] is an extension of the tension and the stiffness energies. In their method, regularization energies for spacing and curvature are applied in tangential and normal directions, respectively, as many has been done in previous research endeavors [27], [28]. Though the paper [25] provides only time marching equations, their regularization scheme can be generalized by the energy term,

$$J_R(\mathbf{Q}) = \alpha J_S(\mathbf{Q}) + \beta J_C(\mathbf{Q}). \tag{14}$$

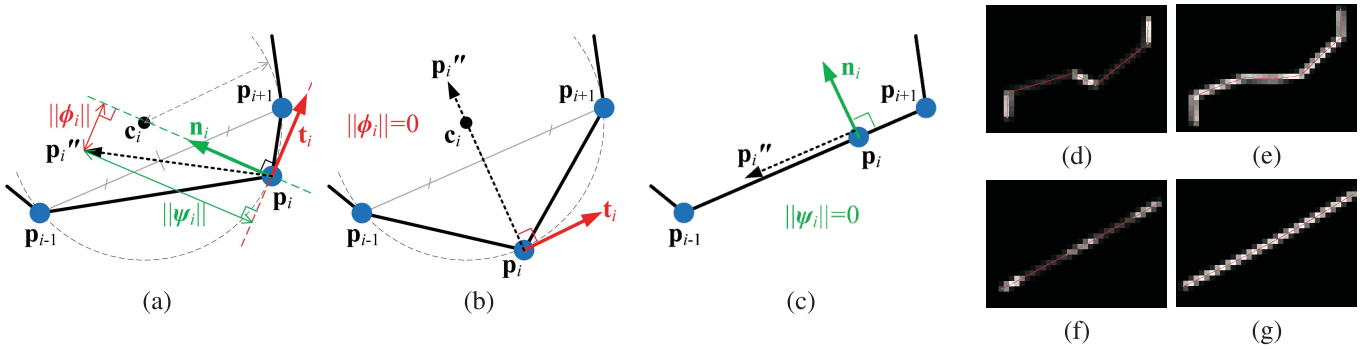


Fig. 4. Regularization of the piecewise-linear model. (a) An example configuration of the control points. (b) $\|\phi_i\| = 0$ when $\mathbf{p}_i$ is equidistant from $\mathbf{p}_{i-1}$ and $\mathbf{p}_{i+1}$. (c) $\|\psi_i\| = 0$ when $\mathbf{p}_i$ lies on the line segment $\overline{\mathbf{p}_{i-1}\mathbf{p}_{i+1}}$. (d) Initial configuration with $N_p = 9$, (e) when $J_S(\mathbf{Q})$ is minimized from (d), (f) when $J_C(\mathbf{Q})$ is minimized from (d), and (g) when both $J_S(\mathbf{Q})$ and $J_C(\mathbf{Q})$ are minimized from (d).

In (14), J_S and J_C represent *spacing regularization* and the *curvature regularization* energies, respectively. The relative importance of the energies is controlled by the parameters, α and β.

The energy terms are defined to reflect the divergences of $\mathbf{Q}$ from the desired configuration. Auxiliary vectors, $\boldsymbol{\phi}_i$ and $\boldsymbol{\psi}_i$ are introduced to quantify the divergences,

$$\boldsymbol{\phi}_i = [p_i''^{(x)} t_i^{(x)}, p_i''^{(y)} t_i^{(y)}]^T, \boldsymbol{\psi}_i = [p_i''^{(x)} n_i^{(x)}, p_i''^{(y)} n_i^{(y)}]^T, \tag{15}$$

where $\mathbf{t}_i = [t_i^{(x)}, t_i^{(y)}]^T$, $\mathbf{n}_i = [n_i^{(x)}, n_i^{(y)}]^T$, and $\mathbf{p}_i'' = [p_i''^{(x)}, p_i''^{(y)}]^T$ are the unit tangential vector, unit normal vector, and second derivative of $\mathbf{C}(s;\mathbf{Q})$, respectively. Geometric interpretation of the vectors are illustrated in Fig. 4(a). While the tangential and the normal vectors are computed from continuous interpolation of $\mathbf{C}(s;\mathbf{Q})$, $\mathbf{p}_i''$ is obtained by the finite difference approximation of $\partial^2 \mathbf{C}(s;\mathbf{Q})/\partial s^2$,

$$\mathbf{p}_i'' = \frac{\mathbf{p}_{i-1} - 2\mathbf{p}_i + \mathbf{p}_{i+1}}{\left(1/\left(N_p - 1\right)\right)^2}.$$

For consistency of $\mathbf{p}_i''$ w.r.t. N_p, the domain $s \in [0, N_p - 1]$ is normalized to $[0, 1]$, and hence, grid spacing in the approximation is set to be $1/(N_p - 1)$. By definition, $\mathbf{p}_i''$ always directs to the middle point of $\overline{\mathbf{p}_{i-1}\mathbf{p}_{i+1}}$ from $\mathbf{p}_i$.

The vectors $\mathbf{t}_i$ and $\mathbf{n}_i$ in (15) are determined by approximating $\mathbf{C}(s;\mathbf{Q})$ with a set of circular arcs connecting $\mathbf{p}_i$, as shown in Fig. 4(a). The circumcenter $\mathbf{c}_i$ of $\mathbf{p}_j$, $j \in \{i-1, i, i+1\}$ is found by locating an equidistant point from $\mathbf{p}_j$. From $\mathbf{p}_i$, a unit vector directed towards $\mathbf{c}_i$ is defined as $\mathbf{n}_i$, and $\mathbf{t}_i$ is obtained by perpendicularly rotating $\mathbf{n}_i$,

$$\mathbf{n}_i = (\mathbf{c}_i - \mathbf{p}_i)/\|\mathbf{c}_i - \mathbf{p}_i\|, \quad \mathbf{t}_i = \left[-n_i^{(y)}, n_i^{(x)}\right]^T.$$

The vectors, $\mathbf{n}_i$ and $\mathbf{t}_i$ are normalized to unit lengths to prevent near-zero or near-infinite values in the following steps.

Note that $\|\boldsymbol{\phi}_i\|$ and $\|\boldsymbol{\psi}_i\|$ are equivalent to the norms of the inner products,

$$\|\boldsymbol{\phi}_i\| = \|\mathbf{p}_i'' \cdot \mathbf{t}_i\|, \quad \|\boldsymbol{\psi}_i\| = \|\mathbf{p}_i'' \cdot \mathbf{n}_i\|. \tag{16}$$

The examples in Fig. 4 depict useful properties of $\|\boldsymbol{\phi}_i\|$ and $\|\boldsymbol{\psi}_i\|$ in (16). As shown in Fig. 4(b), $\|\boldsymbol{\phi}_i\|$ vanishes when $\mathbf{p}_i$ is equidistant from $\mathbf{p}_{i-1}$ and $\mathbf{p}_{i+1}$, which is the case in uniform parametrization [24]. On the other hand, when $\mathbf{p}_i$ lies on the line segment $\overline{\mathbf{p}_{i-1}\mathbf{p}_{i+1}}$, the lowest curvature is achieved and $\|\boldsymbol{\psi}_i\| = 0$ as shown in Fig. 4(c). Based on the properties, $J_S(\mathbf{Q})$ and $J_C(\mathbf{Q})$ are defined as the squared sums of $\|\boldsymbol{\phi}_i\|$ and $\|\boldsymbol{\psi}_i\|$,

$$J_S(\mathbf{Q}) = \frac{1}{2}\sum_{i=0}^{N_p-1} \|\boldsymbol{\phi}_i\|^2, \quad J_C(\mathbf{Q}) = \frac{1}{2}\sum_{i=0}^{N_p-1} \|\boldsymbol{\psi}_i\|^2. \tag{17}$$

Fig. 4 (d)–(f) demonstrate the effects of $J_S(\mathbf{Q})$ and $J_C(\mathbf{Q})$ on $k(\mathbf{x};\mathbf{Q})$. Minimizing J_C preserves the relative spacing of the control points, and the overall curvature of $\mathbf{C}(s;\mathbf{Q})$ is unchanged during the minimization of J_S. Note that J_S contributes not only to equalizing the control point spacing but also to equalizing the intensities of $k(\mathbf{x};\mathbf{Q})$ as can be seen in Fig. 4(e) and (g). To summarize, $J_R(\mathbf{Q})$ promotes elongated and equally-spaced configuration of control points to form a curvy shape for $k(\mathbf{x};\mathbf{Q})$, rather than a broadly spread out one, which results in a non-sparse kernel.

For notational convenience, a vector composed of the second derivatives, $\mathbf{p}_i'' = [p_i''^{(x)}, p_i''^{(y)}]^T$, is expressed by the multiplication of $\mathbf{Q}$ with the second derivative operator Γ,

$$\Gamma\mathbf{Q} = \left[\ldots, p_i''^{(x)}, p_i''^{(y)}, \ldots\right]^T, \ i = 0, \ldots N_p - 1. \tag{18}$$

Combining (18) with (16) and (17), the proposed regularization energy $J_R(\mathbf{Q})$ in (14) can be rewritten in a matrix form,

$$J_R(\mathbf{Q}) = \frac{1}{2}\mathbf{Q}^T\Gamma^T(\alpha\mathbf{T} + \beta\mathbf{N})\Gamma\mathbf{Q}, \tag{19}$$

where $\mathbf{T} = \text{diag}(\ldots, \|t_i^{(x)}\|^2, \|t_i^{(y)}\|^2, \ldots)$ and $\mathbf{N} = \text{diag}(\ldots, \|n_i^{(x)}\|^2, \|n_i^{(y)}\|^2, \ldots)$ are $2N_p \times 2N_p$ diagonal matrices.

B. Optimization of the Energy Function

Under the continuous optimization framework, $J_K\left(\tilde{f}, \mathbf{Q}\right)$ in (5) is minimized by the Levenberg-Maquardt modification of the Gauss-Newton method. The method mitigates difficulties in controlling time step of the gradient descent and reduces the instability of the conventional Newton's method [29].

The update of $\mathbf{Q}_\tau$ at the τ-th step is performed as in (20).

$$\mathbf{Q}_{\tau+1} = \mathbf{Q}_\tau - \left(\widetilde{\mathbf{H}}_K\left(\mathbf{Q}_\tau\right) + \mu_\tau \mathbf{I}\right)^{-1} \nabla J_K\left(\mathbf{Q}_\tau\right), \quad (20)$$

where ∇J_K and $\widetilde{\mathbf{H}}_K$ denote a gradient and an approximated Hessian matrix of $J_K(\mathbf{Q})$, respectively. In (20), the Levenberg-Maquardt method is applied in order to improve the stability of the estimation, by adding a weighted identity matrix, $\mu_\tau \mathbf{I}$. $\nabla J_K(\mathbf{Q})$ and $\widetilde{\mathbf{H}}_K(\mathbf{Q})$ can be sub-divided to further clarify the contributions of $J_D(\mathbf{Q})$ and $J_R(\mathbf{Q})$, such that,

$$\nabla J_K = \nabla J_D + \nabla J_R, \quad \widetilde{\mathbf{H}}_K = \widetilde{\mathbf{H}}_D + \widetilde{\mathbf{H}}_R, \quad (21)$$

where the subscripts D and R represent the relationship with the data and the regularization terms, respectively. From (19), ∇J_R and $\widetilde{\mathbf{H}}_R$ are obtained by fixing $\mathbf{T}$ and $\mathbf{N}$ temporarily,

$$\begin{aligned} \nabla J_R\left(\mathbf{Q}_\tau\right) &= \Gamma^T(\alpha \mathbf{T}_\tau + \beta \mathbf{N}_\tau)\Gamma \mathbf{Q}_\tau, \\ \widetilde{\mathbf{H}}_R\left(\mathbf{Q}_\tau\right) &= \Gamma^T(\alpha \mathbf{T}_\tau + \beta \mathbf{N}_\tau)\Gamma. \end{aligned} \quad (22)$$

Since $J_D(\mathbf{Q})$ is a squared sum of $r(\mathbf{x};\mathbf{Q})$ as in (7),

$$\nabla J_D\left(\mathbf{Q}_\tau\right) = \sum\nolimits_{\mathbf{x}\in\Omega_d} r\left(\mathbf{x};\mathbf{Q}_\tau\right) \nabla r\left(\mathbf{x};\mathbf{Q}_\tau\right), \quad (23)$$

where $\nabla r(\mathbf{x};\mathbf{Q})$ is a $2N_p$ column vector,

$$\nabla r\left(\mathbf{x};\mathbf{Q}_\tau\right) = \left[\ldots, \frac{\partial r\left(\mathbf{x};\mathbf{Q}_\tau\right)}{\partial p_i^{(x)}}, \frac{\partial r\left(\mathbf{x};\mathbf{Q}_\tau\right)}{\partial p_i^{(y)}}, \ldots\right]^T, \quad (24)$$

$i = 0, \ldots, N_p - 1$. The residual $r(\mathbf{x};\mathbf{Q})$ is assumed to be a linear function of $\mathbf{Q}$ in the Gauss-Newton method, and thus the second derivatives of $r(\mathbf{x};\mathbf{Q})$ w.r.t. $\mathbf{p}_i$ are ignored [29]. Hence,

$$\widetilde{\mathbf{H}}_D\left(\mathbf{Q}_\tau\right) = \sum\nolimits_{\mathbf{x}\in\Omega_d} \nabla r\left(\mathbf{x};\mathbf{Q}_\tau\right) \nabla r\left(\mathbf{x};\mathbf{Q}_\tau\right)^T. \quad (25)$$

Since $\widetilde{\mathbf{H}}_K$ is not guaranteed to be positive definite, occasional divergences in $\mathbf{Q}$ have been observed in our experiments without the Levenberg-Maquardt modifications. The parameter μ_τ in (20) determines the degree of compromise between convergence speed and stability, as a larger μ_τ implies a slower but more stable convergence of $\mathbf{Q}$, and vice versa. To minimize potential divergence without sacrificing convergence speed, μ_τ is chosen to be the minimum value that makes $(\widetilde{\mathbf{H}}_K(\mathbf{Q}_\tau)+\mu_\tau \mathbf{I})$ positive definite. Based on the fact that a symmetric matrix is positive definite if it is diagonally dominant and has positive diagonal entries, the optimal μ_τ can be defined as

$$\mu_\tau = \max_i \left\{0, \max_j \left\{\sum\nolimits_{j\neq i} |H_{ij}| - H_{ii}\right\}\right\} + \epsilon, \quad (26)$$

where H_{ij} is the (i, j)-th component of $\widetilde{\mathbf{H}}_K$ and ϵ is a small positive constant. The prescribed procedure for estimating $k(\mathbf{x};\mathbf{Q})$ is summarized in Algorithm 2.

For the optimization of $J_K(\mathbf{Q})$, $\partial r/\partial p_i^{(x)}$ and $\partial r/\partial p_i^{(y)}$ in (24) and (25) are computed using the properties of convolution

$$\begin{aligned} \partial r\left(\mathbf{x};\mathbf{Q}\right)/\partial p_i^{(x)} &= f\left(\mathbf{x}\right) \otimes k_i^{(x)}\left(\mathbf{x};\mathbf{Q}\right) \\ \partial r\left(\mathbf{x};\mathbf{Q}\right)/\partial p_i^{(y)} &= f\left(\mathbf{x}\right) \otimes k_i^{(y)}\left(\mathbf{x};\mathbf{Q}\right) \end{aligned} \quad (27)$$

where $k_i^{(x)}(\mathbf{x};\mathbf{Q}) = \partial k(\mathbf{x};\mathbf{Q})/\partial p_i^{(x)}$. A useful fact for computing (27) is that, only the i-th and the $(i-1)$-th line segments of $\mathbf{C}(s;\mathbf{Q})$, which are connected to $\mathbf{p}_i$, are relevant to $k_i^{(x)}(\mathbf{x};\mathbf{Q})$ and $k_i^{(y)}(\mathbf{x};\mathbf{Q})$. Therefore,

$$\begin{aligned} k_i^{(x)}\left(\mathbf{x};\mathbf{Q}\right) &= \partial k\left(\mathbf{x};\mathbf{Q}\right)/\partial p_i^{(x)} \\ &= \mathcal{D}\Big\{ \int_{s_{i-1}}^{s_i} (\rho - s_{i-1})\, \delta\left(\boldsymbol{\chi} - \mathbf{l}\left(\rho;\mathbf{p}_{i-1},\mathbf{p}_i\right)\right) \mathrm{d}\rho \\ &\quad + \int_{s_i}^{s_{i+1}} (s_{i+1} - \rho)\, \delta\left(\boldsymbol{\chi} - \mathbf{l}\left(\rho;\mathbf{p}_i,\mathbf{p}_{i+1}\right)\right) \mathrm{d}\rho \Big\} \\ &\quad \otimes d^{(x)}\left(\mathbf{x}\right), \end{aligned} \quad (28)$$

where $d^{(x)}(\mathbf{x})$ is a differentiation kernel for the x- direction, e.g. $d^{(x)}(\mathbf{x}) = [1/2, 0, -1/2]$. A definition of $k_i^{(y)}(\mathbf{x};\mathbf{Q})$ is analogous to (28). The computational complexity of (23), (25), and (27) can be significantly reduced via frequency domain processing, i.e. $\mathcal{DFT}\{r(\mathbf{x};\mathbf{Q})\} = \{\mathcal{DFT}\{f\}\mathcal{DFT}\{k\} - \mathcal{DFT}\{b\}\}$. To reduce the computational burdens of DFT itself, methods for directly computing $\mathcal{DFT}\{k\}$, $\mathcal{DFT}\{k_i^{(x)}\}$ are presented in the Appendix.

Algorithm 2 Estimation of $k(\mathbf{x};\mathbf{Q})$, Given $\mathbf{Q}_0$

$\tau = 0$
repeat
  $k(\mathbf{x};\mathbf{Q}_\tau) = \mathcal{D}\{\mathcal{P}\{\mathbf{C}(s,\mathbf{Q}_\tau)\}\}$
  Compute $\partial r(\mathbf{x};\mathbf{Q}_\tau)/\partial p_i^{(x)}$ and $\partial r(\mathbf{x};\mathbf{Q}_\tau)/\partial p_i^{(y)}$ by (27)
  Compute $\nabla J_D(\mathbf{Q}_\tau)$ and $\widetilde{\mathbf{H}}_D(\mathbf{Q}_\tau)$ by (23) and (25)
  Compute $\nabla J_R(\mathbf{Q}_\tau)$ and $\widetilde{\mathbf{H}}_R(\mathbf{Q}_\tau)$ by (22)
  Compute $\nabla J_K(\mathbf{Q}_\tau)$ and $\widetilde{\mathbf{H}}_K(\mathbf{Q}_\tau)$ by (21)
  Compute μ_τ by (26)
  $\mathbf{Q}_{\tau+1} = \mathbf{Q}_\tau - (\widetilde{\mathbf{H}}_K(\mathbf{Q}_\tau) + \mu_\tau \mathbf{I})^{-1} \nabla J_K(\mathbf{Q}_\tau)$
  $\tau = \tau + 1$
until $\tau \geq N_{iter}$ or $\|\mathbf{Q}_{\tau+1} - \mathbf{Q}_\tau\| < threshold$
return $k(\mathbf{x};\mathbf{Q}_\tau)$

C. Coarse-to-Fine Approach

It is common to perform deblurring in a coarse-to-fine fashion [4]–[8] in order to cover large blur kernels. In that regards, a procedure of coarse-to-fine deblurring based on the proposed model is provided in Algorithm 3. At each level, the estimated parameters, $\hat{\mathbf{Q}}$ and $\hat{f}$ are used as the initial parameters for the next finer level. At the coarsest level, in which no initial parameters exist, the $\mathbf{Q}_{init}$ are set by exhaustively searching for a straight line segment which achieves the minimum $J_D(f_{init}, \mathbf{Q})$.

The number of control points, N_p, plays an important role in the kernel estimation, as it controls a balance between flexibility and simplification of the resulting kernel. Also, assigning an excessively large number to N_p can result in the failure of the estimation as a coarse level image provides fewer amount of data observations. Thus, in our implementation, N_p are set to be fairly small, e.g. $N_p = 3$, at the coarsest level and roughly doubled as it goes to the finer levels. According to our experiences, setting N_p to be approximately 20 for the finest level is sufficient to cover most blur cases. Exemplary specifications of N_p are provided in Algorithm 3 and experimental setups in the next section.

Fig. 5 shows the estimated blur kernels using the proposed methods. In this experiment, $f(\mathbf{x})$ and $b(\mathbf{x})$ are given as in

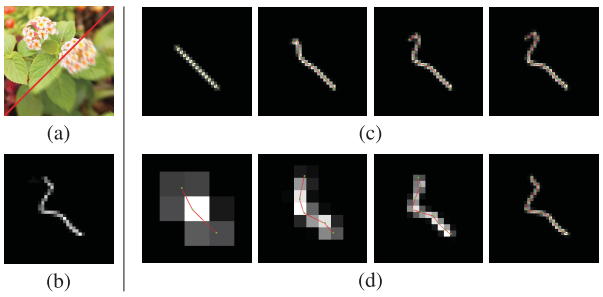


Fig. 5. Estimation of the piecewise-linear model. (a) Latent/blurred images, (b) synthetic kernel applied to (a), (c) kernel estimation by single-level approach, and (d) kernel estimation by coarse-to-fine approach with 4-levels. In (d), N_p is set to be 3, 5, 9, and 17, from the coarsest to the finest levels.

Algorithm 3 Coarse-To-Fine Deblurring

Require: N_L : number of levels, $l = N_L$ denotes the coarsest.
1: Set $N_p = 3$
2: **for** $l = N_L$ to 1 **do**
3: **if** $l = N_L$ **then**
4: $\mathbf{Q}_{init}(N_L) = ExhaustiveSearch(f_{init}(N_L), b(N_L))$
5: **else**
6: $N_p = 2N_p$
7: $\{f_{init}(l), \mathbf{Q}_{init}(l)\} = UpSample(\hat{f}(l+1), \hat{\mathbf{Q}}(l+1))$
8: **end if**
9: $\{\hat{f}(l), \hat{\mathbf{Q}}(l)\} = SingleLevelDeblur\,(f_{init}(l), \mathbf{Q}_{init}(l), b(l))$
10: **end for**
11: **return** $\hat{f}(1)$ and $\hat{\mathbf{Q}}(1)$

Fig. 5(a). As shown in Fig. 5(c) and (d), both of the kernels estimated by the single-level and the coarse-to-fine approaches are quite close to the true kernel shown in Fig. 5 (b). However, the single-level approach suffers because of the local minima as observed in the small stem which branches out at the end of the curve. In the coarse-to-fine approach, the local minima problem is effectively avoided.

IV. Experiment Results

A. Experiments with Synthetically Blurred images

To evaluate the robustness of the proposed blur kernel model and its impact on the quality of the deblurred images, a pilot problem was designed. A set of blurred images was synthesized by convolving the blur kernels and the latent images shown in Fig. 6. The blur kernels were then estimated based on the pairs of the latent and blurred images, and used for deconvolving the blurred images. Noisy image pairs were also generated using the noise model of Boracchi and Foi [1] with a quantum efficiency parameter $\lambda = 1500$ for Poisson noise and $\sigma = 3.0$ for Gaussian noise $\mathcal{N}\left(0, \sigma^2\right)$.

The blur kernel models to be evaluated are: 1) the generic model in (2), 2) the linear model, 3) the piecewise-linear model by Patanukhom and Nishihara [16], and 4) the proposed model. For the generic model, the conjugate gradient method was applied to minimize the energy function, which consists of quadratic data energy similar to (7) and the sparsity energy from (13) with $\lambda_H = 5$ and $\gamma_H = 2$. For the proposed model, the energy function in (5) with $\alpha = 0.03$ and $\beta = 0.005$ was

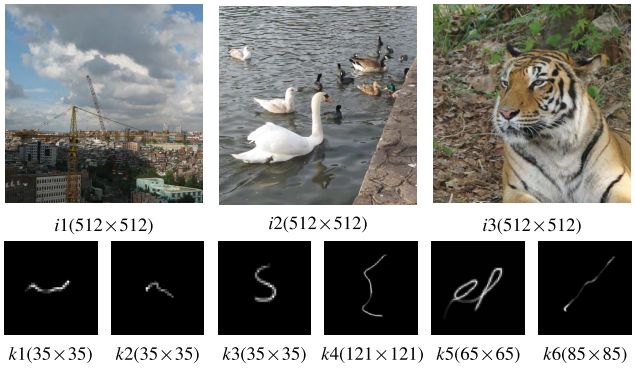


Fig. 6. Set of images and blur kernels used for the experiment. Images and kernels are numbered by $i1 \sim i3$ and $k1 \sim k6$, respectively. Sizes of the images and kernels are annotated, respectively.

TABLE I
Camera Setup for the Images in Fig. 9

Scene	Blurred	Flash	Noisy
bookshelf, f/13	1/10s, ISO320	1/100s, ISO200	1/200s, ISO1600
donkey, f/18	1/5s, ISO400	1/100s, ISO800	1/200s, ISO3200
bench, f/2.8	1/10s, ISO800	1/100s, ISO800	1/200s, ISO3200

employed. Throughout the experiments with 4 image levels, we set $N_p = 21$ at the finest level and 11, 5, 3 for the subsequent coarser ones. The linear model was treated as a special case of the piecewise-linear model with $N_p = 2$, and no regularization energy was involved. For the method of Patanukhom and Nishihara [16], we utilized their blind blur identification algorithm, which employs no regularization.

In the deconvolution stage, the $L1$ and the $L2$ norm regularization energies in (8), and the Richardson-Lucy(RL) deconvolution [30], [31] were tested. The $L1$ and $L2$ norms for image regularization were realized by setting $\gamma_F = 1$ and $\gamma_F = 2$ in (8), respectively. The regularization weights were set to be $\lambda_F = 0.01$ for the image pairs without noise, and $\lambda_F = 0.05$ for the noisy image pairs. For the $L1$ and the $L2$ norm regularizations, a method proposed by Krishnan and Fergus [23] and the direct solver method [8] were used, repectively.

Quantitative evaluation with and without additional noise are summarized in Fig. 7. Representative examples of this experiment are also shown in Fig. 8. As can be seen in Figs. 7 and 8, even with the noise, the proposed model accurately estimates the kernels, except for twisted ones, such as $k5$, which is due to its limited flexibility. The results of the linear model and the Patanukhom and Nishihara [16] are far from successful in every case.

Regarding the quality of the deblurred images, the proposed model outperforms the other methods, except in the cases of the kernel $k5$ and when the RL deconvolution is employed for the noisy images. Note that the RL deconvolution only assumes the Poisson distribution of noise, though the Gaussian noise is also present in the corrupted images. However, as shown in Fig. 8(b) and (c), more image details are preserved

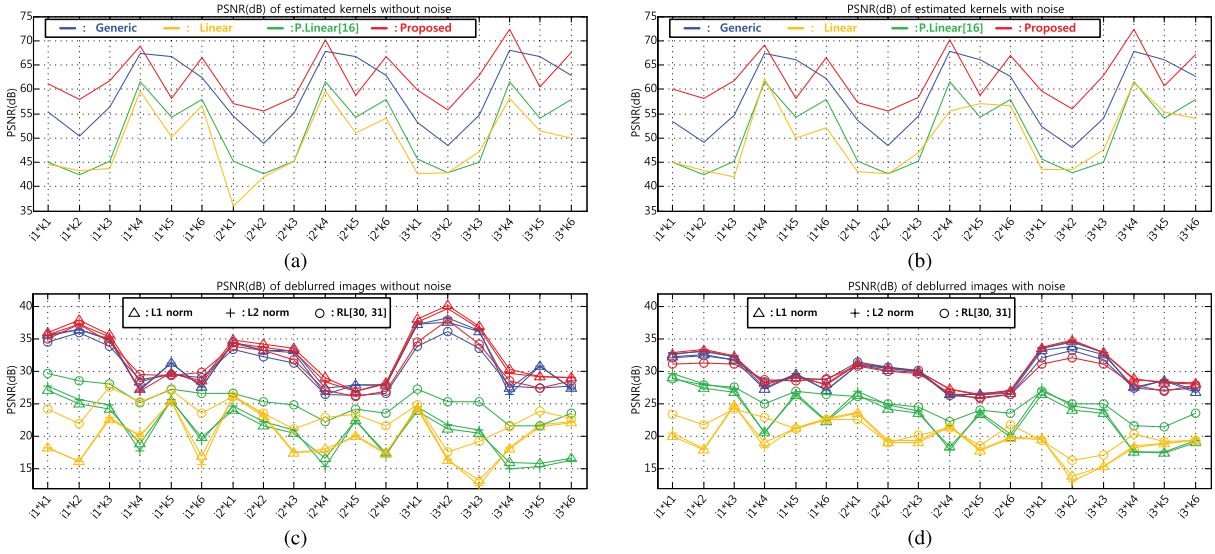


Fig. 7. PSNR values of the blur kernel estimation and the deblurring for Fig. 6. PSNR values of estimated blur kernels (a) without added noise, and (b) with noise [1]. PSNR values of deconvolved images (c) without the noise, and (d) with the noise. Horizontal axes represent combinations of images and blur kernels shown in Fig. 6. The colors and the markers of the lines represent the blur kernel model and the deconvolution methods, respectively.

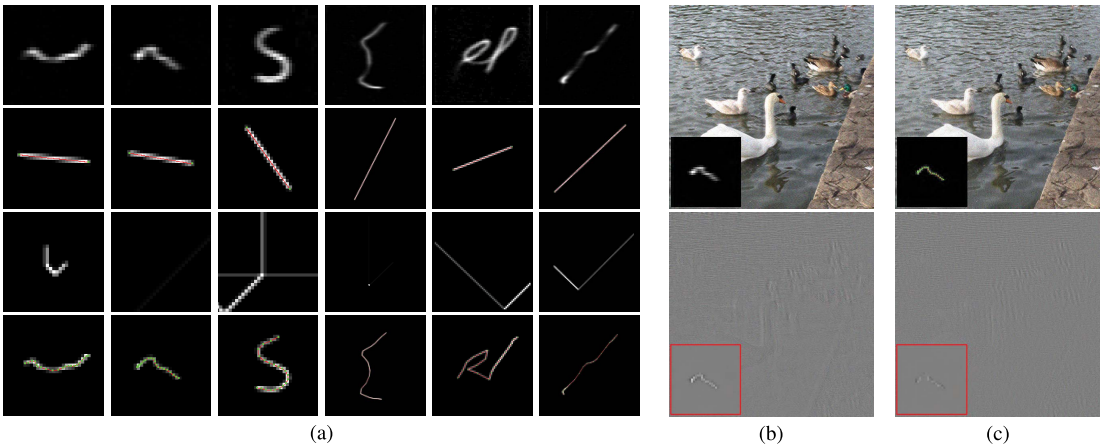


Fig. 8. Selected examples of the experiment results of $i2$ in Fig. 6 with added noise [1]. (a) Estimated blur kernels with noise, from top to bottom, the generic model, the linear model, the piecewise-linear model [16], and the proposed model, (b) results of the generic model(top) and the difference images(bottom), and (c) results of the proposed model(top) and the difference images(bottom). In (a), each column corresponds to $k1\sim k6$ in Fig. 6. The results in (b) and (c) are obtained using $i2$, $k2$ and RL deconvolution [30], [31].

by the proposed model than the generic model, thanks to the robust kernel estimation.

In this experiment, ∇J_D and $\widetilde{\mathbf{H}}_D$ are computed in the frequency domain for enhanced computational efficiency. As such, two methods of computing the $\mathcal{DFT}\{k\,(\mathbf{x};\mathbf{Q})\}$ and $\mathcal{DFT}\{k_i^{(x)}\,(\mathbf{x};\mathbf{Q})\}$ were tested: the first one applies DFT to $k\,(\mathbf{x};\mathbf{Q})$ and $k_i^{(x)}\,(\mathbf{x};\mathbf{Q})$, and the second one is a direct application of (37) and (39) from the Appendix. Our experiment results demonstrate that the second method is roughly 3 times more efficient than the first one, which takes about 100 sec. for the kernel estimation.

B. Experiments With Real-World Blurred Images

To further evaluate the effectiveness of the proposed model on real-world images, several deblurring methods based on the generic model in (2) were implemented and their kernels were replaced by the piecewise-linear ones. In our experimentation with the 4-level coarse-to-fine approach, N_p was set to be 3, 5, 9, and 17, from the coarsest to the finest levels. Note that the values are smaller than those of subsection IV.A, as real-world blurs in our experiment are simpler than the hand-drawn ones in Fig. 6. The images taken by a Nikon D300 are shown in Fig. 9 and the corresponding camera parameters are listed in Table I. As shown in the table, the ISO value and exposure time parameters are set differently to cover various levels of blur and the time-independent and the time-dependent noise components [1] in real-world. Note that no artificial noise is added. The methods involved in the comparison are those proposed by Yuan *et al.* [6], Zhuo *et al.* [7], and Cho and Lee [8]. Table II summarizes the methods in terms of the input image configuration, kernel estimation, and deconvolution.

TABLE II
SUMMARY OF DEBLURRING METHODS IN COMPARISON

	Yuan *et al.* [6]	Zhuo *et al.* [7]	Cho and Lee [8]
Input images	Blurred+noisy	Blurred+flash	Blurred only
Kernel estimation	Gradient descent	IRLS [32], [33]	Conjugate gradient
Kernel regularization in (13)	$L2$ sparsity, $\gamma_H = 2$	$L0.8$ sparsity, $\gamma_H = 0.8$	$L2$ sparsity, $\gamma_H = 2$
Deconvolution	Modified RL [30], [31]	IRLS [32], [33]	Direct solver $*$
Image Regularization in (8)	None	$L1$ norm of ∇f, $\gamma_F = 1$	$L2$ norm of ∇f, $\gamma_F = 2$ $*$

$*$ Note that, in [8], a deconvolution method of Shan *et al.* [5] is used for obtaining the final deblurred image.

Fig. 9. Set of images for the experiments. From top to bottom:, blurred, flash, and noisy images respectively. The images are of size 1024×1024. Sizes of kernels for the generic model are set to be 75×75, 85×85, and 85×85, respectively.

In Yuan *et al.* [6], the kernel estimated from a blurred/noisy image pair is used for the deconvolution. In the experiment, the post-processing stage, which involves de-ringing and the addition of image details, was omitted to focus on the accuracy of the kernel estimation. As seen in Fig. 10, the generic model is easily corrupted by noise. The estimated kernels essentially become thicker and randomly distributed dots arise with greater frequency. In contrast, the piecewise-linear blur kernels remain thin due to their low DOF and inherent sparsity. In regards to image quality, the proposed blur kernel maintains most of the image details.

The piecewise-linear model has a greater impact on the algorithms of Zhuo *et al.* [7] and Cho and Lee [8], since the estimated kernel is recursively fed to the next deconvolution stage. As demonstrated in Fig. 11, the blur kernels represented by the generic model in (2) suffer from noise. Even with carefully adjusted parameters, obtaining a clear kernel for the scene *bench* is impossible. While not perfect, the blur kernels estimated using the piecewise-blur model are more acceptable,

TABLE III
COMPUTATIONAL TIME OF DEBLURRING METHODS IN COMPARISON

Kernel representation	Processing stage	Yuan *et al.* [6]	Zhuo *et al.* [7]	Cho and Lee [8]
Generic model	Kernel estimation	29.80s	393.48s	32.29s
	Deconvolution	52.64s	264.13s	2.46s
	Total	82.57s	1010.60s	70.10s
Proposed, using explicit DFT	Kernel estimation	639.89s	69.21s	162.29s
	Deconvolution	52.67s	353.24s	3.17s
	Total	692.56s	777.85s	207.77s
Proposed, using (37)&(39) in Appendix	Kernel estimation	87.05s	17.35s	49.77s
	Deconvolution	53.13s	352.09s	3.24s
	Total	140.18s	723.25s	96.11s

TABLE IV
PSNR VALUES OF CHO AND LEE [8] WITH THE PIECEWISE-LINEAR MODEL FOR THE BENCHMARK DATA SET [22]

Blur no.	*Image*1	*Image*2	*Image*3	*Image*4
*Blur*1 $(41 \times 41)*$	**34.438**	**29.633**	**36.263**	31.280
*Blur*2 (41×41)	31.347	25.086	30.529	26.047
*Blur*3 $(41 \times 41)*$	**33.594**	**30.106**	**34.687**	**31.232**
*Blur*4 $(41 \times 41)*$	**36.557**	**29.449**	**33.749**	**31.765**
*Blur*5 $(41 \times 41)*$	29.502	25.519	**34.680**	**34.338**
*Blur*6 (41×41)	**31.379**	25.368	31.871	25.598
*Blur*7 (41×141)	30.547	24.219	30.483	27.381
*Blur*8 $(49 \times 141)*$	**25.579**	20.475	**25.245**	**21.664**
*Blur*9 $(141 \times 141)*$	27.561	22.392	25.821	25.569
*Blur*10 (141×141)	24.640	17.250	24.213	18.455
*Blur*11 $(74 \times 139)*$	**27.317**	22.512	**27.562**	**22.650**
*Blur*12 (41×41)	**31.517**	**27.001**	31.468	**27.355**

and the deblurred images are clearer than those using the generic model.

The piecewise-linear model adopted to the method by Cho and Lee [8] drastically improves the accuracy of the kernel estimation and the quality of the deblurred images. In their method, the kernel estimation and the image deconvolution are alternately applied, as they were in other blind deblurring methods [4], [5]. The results in Fig. 12 show that the entire deblurring process is severely affected by the noise included in the images. To demonstrate the effects of the estimated kernels more clearly, the images shown in Fig. 12 are obtained by using a deconvolution with the $L2$ norm of image gradients, which is used in the kernel estimation stage of Cho and Lee [8]. As the noise level increases, the generic model rapidly loses information on the camera's motion, while the proposed model maintains robustness, thanks to its low DOF.

The average computational time of the deblurring methods in [6]–[8] for the images shown in Fig. 9 are listed in Table III. As shown in the table, the kernel estimation based on the generic model with $L2$ sparsity runs faster than the case with the proposed model. Since $k(\mathbf{x}; \mathbf{Q})$ is manipulated as a concatenation of $k_i(\mathbf{x}; \mathbf{Q})$, $i = 0, \ldots, N_p - 2$, computational time for the kernel estimation is also increased by a factor of $(N_p - 1)$. However, the method introduced in the Appendix significantly shortens the time, as can be seen in the last three rows of Table III. Note that the non-linear nature of the $L0.8$ norm, used in Zhuo *et al.* [7], leads to an exceptionally high computational cost for the generic model, compared to the cases of the generic model

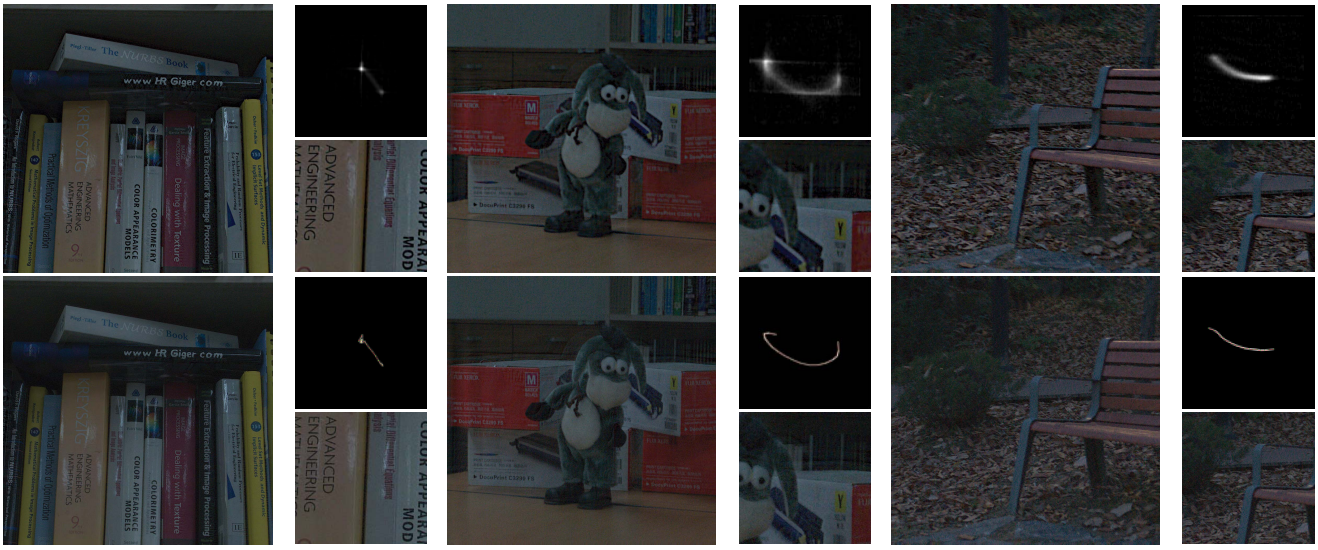


Fig. 10. Results of Yuan *et al.* [6] using the generic (top row) and the proposed (bottom row) representation for blur kernels.

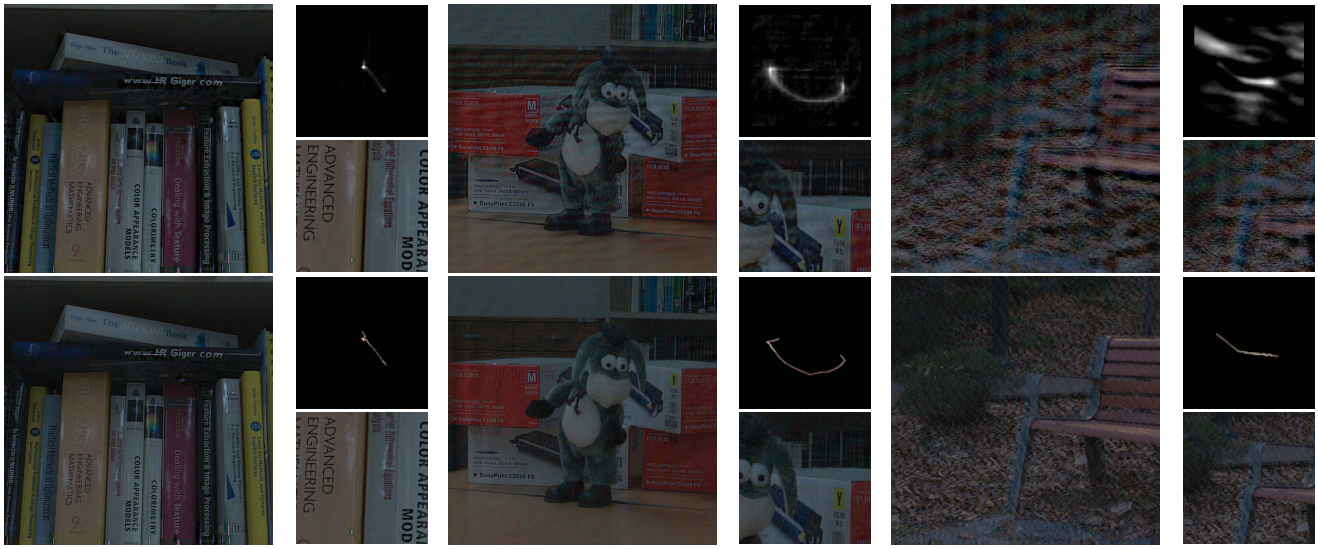


Fig. 11. Results of Zhuo *et al.* [7] using the generic (top row) and the proposed (bottom row) representation for blur kernels.

with the $L2$-norm regularization [6], [8] and the proposed model.

C. Experiments With a Benchmark Data Set [22]

In a recent report, Köhler *et al.* [22] provided a list of the PSNR values of blind deblurring algorithms for images of their benchmark data set with various degrees of blurring. The data set includes not only uniform but also non-uniform blurs which are more realistic than ones we assumed in (1). We evaluated the blind deblurring, which is based on the scheme of Cho and Lee [8] but instead adopting the proposed model. PSNRs of the deblurred image are listed in Table IV. In the table, approximately uniform blurs are marked by an asterisk (*), and the approximate sizes of the kernels are annotated in the leftmost pane. Furthermore, PSNRs higher than those of Cho and Lee [8] and Whyte *et al.* [19] are highlighted by bold numbers and underlines, respectively. Refer to [22] for details on the data set and the performance of the existing blind deblurring methods.

According to the result, the overall performance of the piecewise-linear model is comparable to that of Cho and Lee [8], except in the case of non-uniform blurs. For such cases, which violates the assumption of uniform blur in (1), the flexibility of the generic model is found to be more advantageous than the simplicity of the proposed model. It must be noted that the blind deblurring method (Cho and Lee [8] and the piecewise-linear model in conjunction) performed better than the ones designed for non-uniform deblurring [19], [21] when tested on large blur cases, such as on $Blur8 \sim Blur11$, as shown in Fig. 13(b) and (c). This suggests that a model with a low DOF can be a more suitable solution than one with a higher DOF, especially when a large blur is involved.

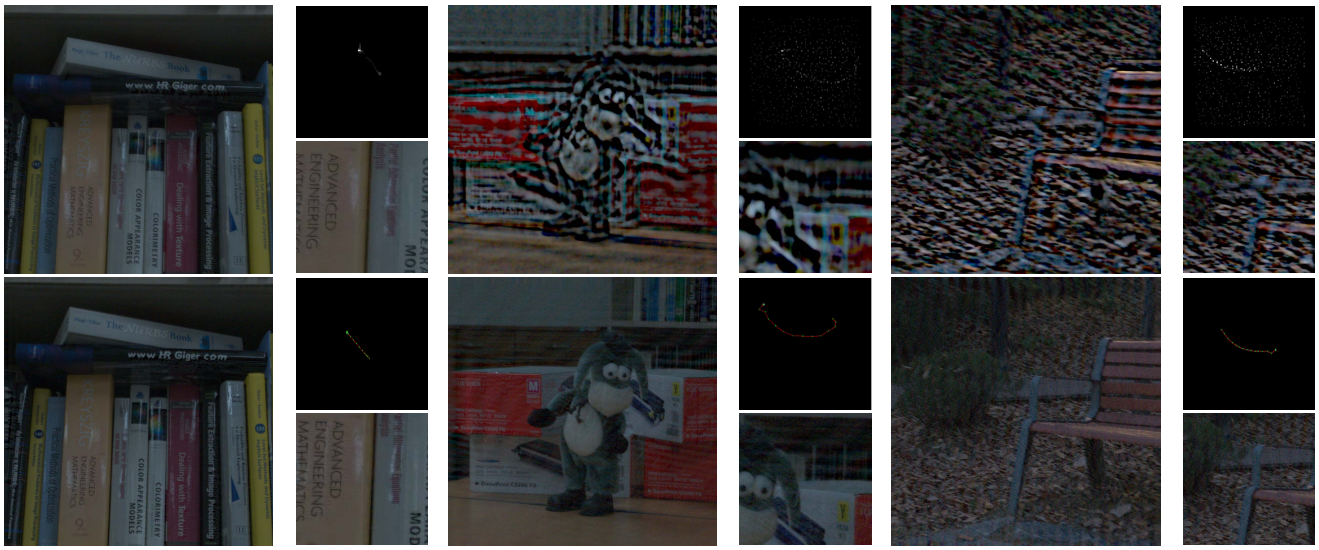


Fig. 12. Results of Cho and Lee [8] using the generic (top row) and the proposed (bottom row) representation for blur kernels.

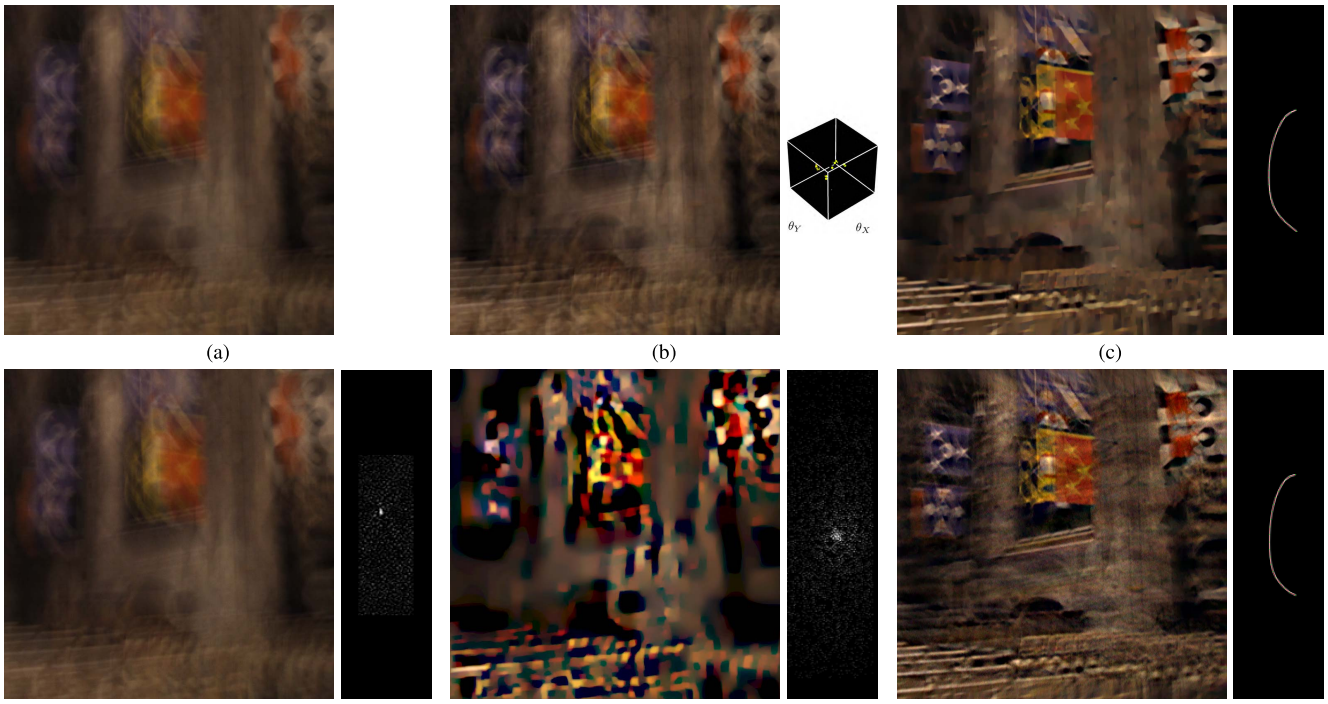


Fig. 13. Experiment results using *Image*1 and *Blur*8 in the benchmark data set [22]. (a) Blurred image, (b) Whyte *et al.* [19], (c) Cho and Lee [8] + proposed model, (d) Xu and Jia [34] with noise, (e) Cho and Lee [8] with noise, and (f) Cho and Lee [8] + proposed model with noise.

To verify the robustness of the proposed method to noise, we conducted a performance comparison of three uniform deblurring methods: Xu and Jia [34], Cho and Lee [8], and Cho and Lee [8] + the piecewise-linear model. Noise similar to the one used in subsection IV.A, but less severe one ($\lambda = 3000$ and $\sigma = 1.5$), were added to the blurred images provided by Köhler *et al.* [22]. The PSNRs of the deblurred images are listed in Table V, and the examples are shown in Fig. 13 (d)–(f). Note that the results of Cho and Lee [8] show more sensitivity to noise than ones using the piecewise-linear model. The deblurring method of Xu and Jia [34], which is also based on the generic model has shown good overall performance on the noisy images, however, failed in dealing with large blurs (*Blur*8∼*Blur*11). As shown in Fig. 13(d) and (e), the generic model is severely corrupted by noise, as opposed to the piecewise-linear model shown in Fig. 13(f). It is worth of noting that Xu and Jia [34] records the best PSNRs in the original benchmark, although uniform blur is assumed in contrast to the non-uniform models [19], [21]. That is, flexible models for large blurs do not necessarily lead to better performance. The same applies to the proposed model, compared to the generic models. Based on these observations,

TABLE V
PSNR VALUES OF XU AND JIA [34], CHO AND LEE [8], AND CHO AND LEE [8] + PIECEWISE-LINEAR MODEL IN THE PRESENCE OF NOISE [1]

	Xu and Jia [34]				Cho and Lee [8]				Cho and Lee [8] with the piecewise-linear model			
Blur no.	*Image*1	*Image*2	*Image*3	*Image*4	*Image*1	*Image*2	*Image*3	*Image*4	*Image*1	*Image*2	*Image*3	*Image*4
*Blur*1 $(41 \times 41)*$	**32.885**	**29.225**	**35.214**	**31.776**	26.384	21.646	21.630	22.923	30.588	26.996	30.057	28.194
*Blur*2 (41×41)	**33.526**	**29.940**	**32.507**	**31.276**	27.321	22.118	22.427	22.076	29.873	24.310	28.066	24.013
*Blur*3 $(41 \times 41)*$	**34.985**	**29.411**	**33.401**	**31.684**	27.239	23.174	23.338	24.791	30.858	26.694	31.047	29.177
*Blur*4 $(41 \times 41)*$	**34.107**	**28.972**	**33.250**	**31.670**	26.196	21.165	23.062	22.513	30.773	27.259	30.288	28.946
*Blur*5 $(41 \times 41)*$	**33.740**	**28.820**	34.389	33.946	25.390	19.308	33.649	33.223	29.123	22.006	**34.632**	**34.308**
*Blur*6 (41×41)	**32.549**	**27.402**	**31.186**	**29.053**	24.543	19.335	19.700	19.563	28.773	23.118	27.177	23.493
*Blur*7 (41×141)	**32.580**	**28.120**	**31.748**	**29.306**	24.252	18.993	20.292	19.264	28.556	23.488	29.116	26.009
*Blur*8 $(49 \times 41)*$	16.877	14.477	17.801	**16.783**	17.681	13.669	15.648	13.880	**24.948**	**16.823**	**24.340**	15.839
*Blur*9 $(141 \times 141)*$	15.550	12.488	13.156	13.087	17.277	11.752	13.844	14.493	**26.175**	**21.586**	**24.642**	**23.519**
*Blur*10 (141×141)	16.195	11.961	14.204	12.222	18.809	12.975	14.022	14.314	**23.756**	**17.636**	**22.098**	**18.578**
*Blur*11 $(74 \times 139)*$	15.351	12.979	12.613	13.240	24.021	17.692	18.830	17.397	**25.687**	**22.110**	**24.455**	**20.899**
*Blur*12 (41×41)	**32.173**	**25.927**	**31.553**	**27.596**	25.318	18.043	18.407	20.110	28.979	25.169	25.266	25.157

we can conclude that the piecewise-linear model is more robust than the generic model, especially when large blurs and noise are involved.

V. CONCLUSION

A novel framework for blur kernel estimation has been implemented. The entire framework, which includes a combination of the generally applicable kernel estimation method with the piecewise-linear model and an effective regularization scheme, is designed to improve the robustness of the estimation process in the presence of noise and large blurs. The results of experiments with various imaging setups and deconvolution methods prove that the estimated kernels are significantly more accurate and robust to noise, compared to the conventional approaches.

In spite of the effectiveness and versatility of the proposed model, there is room for improvement. Efforts for lessening the computational burden, such as the one presented in the Appendix, need to be further pursued. Parallelization of the proposed algorithm using a GPGPU is also being sought for additional improvements in speed. In addition, research on optimization methods and determination of α and β for the regularization terms in (14), which are crucial in obtaining accurate results, is required to mitigate the problems of parameter dependency and handling highly twisted blur kernels.

Based on our experiences, we believe that the proposed method can improve most existing deblurring methods in regards to robustness and accuracy. Moreover, we expect that the proposed model can also be applicable to non-uniform motion deblurring methods [19]–[21].

APPENDIX
DFT OF PIECEWISE-LINEAR BLUR KERNEL

The numerical implementation of (23), (25) and (27) can benefit significantly from the usage of frequency domain computations based on the convolution theorem and Parseval's theorem. Here, a direct computation of $\mathcal{DFT}\{k(\mathbf{x})\}$, $\mathcal{DFT}\{k_i^{(x)}(\mathbf{x})\}$ and $\mathcal{DFT}\{k_i^{(y)}(\mathbf{x})\}$ without the explicit DFT is presented to increase procedural efficiency.

Algorithm 4 Determining Cont. Sub-Intervals, $\{s_1, \ldots, s_{n_i}\}$

$s_1^{(x)} = s_1^{(y)} = 0$
$\mathbf{d}_i = [d_i^{(x)}, d_i^{(y)},]^T = \mathbf{p}_{i+1} - \mathbf{p}_i$
for $* = x$ to $* = y$ **do**
  if $d_i^{(*)} > 0$ **then**
    $q^{(*)} = \lceil p_i^{(*)} \rceil,\ r^{(*)} = 1$
  else
    $q^{(*)} = \lfloor p_i^{(*)} \rfloor,\ r^{(*)} = -1$
  end if
end for
for $* = x$ to $* = y$ **do**
  $j = 1$
  while $s_j^{(*)} < 1$ **do**
    $s_j^{(*)} = |q^{(*)} - p_i^{(*)}| / |d_i^{(*)}|$
    $q^{(*)} = q^{(*)} + r^{(*)},\ j = j + 1$
  end while
end for
$n_i = 1$, $s_1 = 0$, $j, k = 2$
while $s_j^{(x)} \leq 1$ and $s_k^{(y)} \leq 1$ **do**
  $n_i = n_i + 1$
  if $s_j^{(x)} < s_k^{(y)}$ and **then**
    $s_{n_i} = s_j^{(x)},\ j = j + 1$
  else
    $s_{n_i} = s_k^{(y)},\ k = k + 1$
  end if
end while
return $\{s_1, \ldots, s_{n_i}\}$ and n_i

From (9), the $K_i(\mathbf{u}; \mathbf{Q}) = \mathcal{DFT}\{k_i(\mathbf{x}; \mathbf{Q})\}$ can be expressed as (29),

$$\begin{aligned} K_i(\mathbf{u}; \mathbf{Q}) &= \mathcal{DFT}\{\mathcal{D}\{\mathcal{P}\{\mathbf{l}(s; \mathbf{p}_i, \mathbf{p}_{i+1})\}\}\} \\ &= \int_0^1 \Delta(\mathbf{u}; \mathbf{l}(s; \mathbf{p}_i, \mathbf{p}_{i+1}))\, \mathrm{d}s \end{aligned} \tag{29}$$

where $\mathbf{u} = [u, v]^T \in \mathbb{Z}^2$ is a periodic 2-dimensional frequency domain and $\Delta(\mathbf{u}; \mathbf{x}_0) = \mathcal{DFT}\left\{\mathcal{D}\left\{\delta\left(\boldsymbol{\chi} - \boldsymbol{\chi}_0\right)\right\}\right\}$. To properly express $K_i(\mathbf{u}; \mathbf{Q})$, we first obtain $\Delta\left(\mathbf{u}; \boldsymbol{\chi}'\right)$ and then shift and integrate it along a straight line as described in the last row of (29). Note that the bilinear sampling kernel is applied to kernel representation as in (12). Then we have

$$\begin{aligned} \Delta\left(\mathbf{u}; \boldsymbol{\chi}'\right) &= \mathcal{DFT}\left\{\mathcal{D}\left\{\delta\left(\boldsymbol{\chi} - \boldsymbol{\chi}'\right)\right\}\right\} \\ &= G\left(u, \chi'\right) G\left(v, \xi'\right), \end{aligned} \tag{30}$$

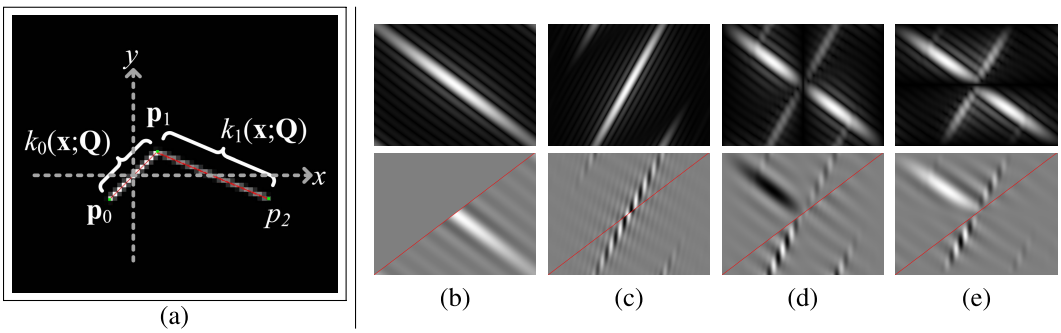


Fig. 14. The piecewise-linear blur kernel in the discrete frequency domain. (a) control points and the corresponding blur kernel in 640×480 discrete spatial domain, (b)-(c) DFT of $k_0(\mathbf{x};\mathbf{Q})$ and $k_1(\mathbf{x};\mathbf{Q})$ which correspond to the first and second line segments shown in (a), and (d)-(f) differentiation of (b) and (c) with $p_1^{(x)}$ and $p_1^{(y)}$, respectively. In (b)-(e), the power spectrum and the real/imaginary parts are shown in the upper and the lower rows, respectively.

where $G\left(u,\chi'\right)=\sum_{u'=-\infty}^{\infty}\mathrm{sinc}^2\left(u-u'\right)e^{-j\frac{2\pi}{M}(u-u')\chi'}$, for an $M\times N$ image. For the convenience of the following descriptions, additional abbreviations are set as

$$\begin{aligned} G\left(u,\chi'\right) &= e^{-j\frac{2\pi}{M}u\chi'}H\left(u,\chi'\right), \\ H\left(u,\chi'\right) &= \sum_{u'=-\infty}^{\infty}\mathrm{sinc}^2\left(u-u'\right)e^{-j\frac{2\pi}{M}u'\chi'}. \end{aligned} \quad (31)$$

Although the infinite summation in (31) is impossible to implement, a discrete time Fourier transform (DTFT) pair,

$$\begin{aligned} \sum_{u'=-\infty}^{\infty} w\mathrm{sinc}^2\left(wu'\right)e^{-j\frac{2\pi}{M}\chi'u'} &= \mathcal{DTFT}\left\{w\mathrm{sinc}^2\left(wu'\right)\right\} \\ &= \mathrm{tri}\left(\frac{\chi'}{w}\right), \quad \chi'\in\left[-\tfrac{1}{2},\tfrac{1}{2}\right), \ w<0.5, \end{aligned} \quad (32)$$

can be found. Since the triangular function $\mathrm{tri}\left(\chi\right)$ is used as a sampling kernel in (12), we set $w=1$ for (32). Note that aliasing has occurred since $w>0.5$. For further simplification, time reversal and time domain shifting are applied. Then, $H\left(u,\chi'\right)$ can be rewritten using a sum of periodic repetition of (32) modulated by complex exponentials,

$$H\left(u,\chi'\right)=\sum_{k=-\infty}^{\infty}\mathrm{tri}\left(\chi'-m\right)e^{j\frac{2\pi}{M}u(\chi'-m)}. \quad (33)$$

Similarly, $G\left(u,\chi'\right)$ can be rewritten using (33).

$$G\left(u,\chi'\right)=\sum_{k=-\infty}^{\infty}\mathrm{tri}\left(\chi'-m\right)e^{-j\frac{2\pi}{M}um} \quad (34)$$

Looking into (34) closely, we find that $G\left(u,\chi'\right)$ is merely a linear interpolation of the sequence $\{\ldots,e^{-j\frac{2\pi}{M}um},\ldots\}$, $m\in\mathbb{Z}$. In other words,

$$G\left(u,\chi'\right)=\left(\lceil\chi'\rceil-\chi'\right)e^{-j\frac{2\pi}{M}u\lfloor\chi'\rfloor}+\left(\chi'-\lfloor\chi'\rfloor\right)e^{-j\frac{2\pi}{M}u\lceil\chi'\rceil}. \quad (35)$$

$G\left(v,\xi'\right)$ is also defined similarly. Note the interesting relation that exists between the sampling theory and (35). In DFT/DTFT, fractional shifting is forbidden and only an integer multiple of the grid size is allowed for time domain shifting. According to (35), the DFT of an arbitrary impulse convolved by the triangular kernel is equivalent to a linear interpolation of the two exponentials which correspond to the integer-amount of time domain shifting.

Using (30) and (35), $K_i\left(\mathbf{u};\mathbf{Q}\right)$ can be rewritten.

$$\begin{aligned} K_i\left(\mathbf{u};\mathbf{Q}\right) = \int_0^1 & G\left(u,l^{(x)}\left(s;\mathbf{p}_i,\mathbf{p}_{i+1}\right)\right) \\ & \cdot G\left(v,l^{(y)}\left(s;\mathbf{p}_i,\mathbf{p}_{i+1}\right)\right)\mathrm{d}s \end{aligned} \quad (36)$$

As we increase the dummy variable s in (36), $G\left(u,l^{(x)}\left(s;\mathbf{p}_i,\mathbf{p}_{i+1}\right)\right)$ and $G\left(v,l^{(y)}\left(s;\mathbf{p}_i,\mathbf{p}_{i+1}\right)\right)$ asynchronously pass through the discontinuous points according to the slope of the line segment $\overline{\mathbf{p}_i\mathbf{p}_{i+1}}$. Therefore, the original interval of integration $[0,1]$ should be divided into continuous sub-intervals. Algorithm 4 shows a pseudo code for determining the sub-intervals.

$$\begin{aligned} K_i\left(\mathbf{u};\mathbf{Q}\right) &= \sum_{j=1}^{n_i-1}\int_{s_j}^{s_{j+1}} G\left(u,l^{(x)}\left(s;\mathbf{p}_i,\mathbf{p}_{i+1}\right)\right) \\ &\quad \cdot G\left(v,l^{(y)}\left(s;\mathbf{p}_i,\mathbf{p}_{i+1}\right)\right)\mathrm{d}s \\ &= \sum_{j=1}^{n_i-1}\int_{s_j}^{s_{j+1}} A_j^{(i)}\left(s\right)\mathrm{d}s \end{aligned} \quad (37)$$

In the second equality of (37), the definite integral of $G\left(u,l^{(x)}\right)G\left(v,l^{(y)}\right)$ within the continuous sub-intervals is substituted with $\int_{s_j}^{s_{j+1}}A_j^{(i)}\left(s\right)\mathrm{d}s$. Because there is no discontinuity in $\left[s_j,s_{j+1}\right]$,

$$\begin{aligned} &\int_{s_j}^{s_{j+1}} A_j^{(i)}\left(s\right)\mathrm{d}s \\ &\qquad = \frac{s_{j+1}-s_j}{6}\left\{a_1\left(2b_1+b_2\right)+a_2\left(b_1+2b_2\right)\right\}, \\ &\text{where } a_1 = G\left(u,l^{(x)}\left(s_j;\mathbf{p}_i,\mathbf{p}_{i+1}\right)\right), \\ &\qquad a_2 = G\left(u,l^{(x)}\left(s_{j+1};\mathbf{p}_i,\mathbf{p}_{i+1}\right)\right), \\ &\qquad b_1 = G\left(v,l^{(y)}\left(s_j;\mathbf{p}_i,\mathbf{p}_{i+1}\right)\right), \\ &\qquad b_2 = G\left(v,l^{(y)}\left(s_{j+1};\mathbf{p}_i,\mathbf{p}_{i+1}\right)\right). \end{aligned} \quad (38)$$

The closed form expression of integral in (38) was found using basic calculus,

$$\begin{aligned} &\int_0^1 \left(\left(1-s\right)a_1+sa_2\right)\left(\left(1-s\right)b_1+sb_2\right)\mathrm{d}s \\ &\qquad = \tfrac{a_1(2b_1+b_2)+a_2(b_1+2b_2)}{6}. \end{aligned}$$

The $K_i\left(\mathbf{u};\mathbf{Q}\right)$ generated by combining (36) and (37) with a control point configuration in Fig. 14(a) is shown in Fig. 14(b) and (c).

Similar to (37) and (38), the DFT of $\partial k(\mathbf{x};\mathbf{Q})/\partial p_i^{(x)}$ for (23), (25) and (27) can be found as follows.

$$\begin{aligned}\mathcal{DFT}&\left\{\frac{\partial k(\mathbf{x};\mathbf{Q})}{\partial p_i^{(x)}}\right\}\\ &=\frac{\partial K(\mathbf{u};\mathbf{Q})}{\partial p_i^{(x)}}=\frac{\partial K_i(\mathbf{u};\mathbf{Q})}{\partial p_i^{(x)}}+\frac{\partial K_{i-1}(\mathbf{u};\mathbf{Q})}{\partial p_i^{(x)}}\\ &=D^{(x)}(\mathbf{u})\sum_{j=1}^{n_i-1}\int_{s_j}^{s_{j+1}}(1-s)A_j^{(i)}(s)\,\mathrm{d}s\\ &\quad+D^{(x)}(\mathbf{u})\sum_{j=1}^{n_{i-1}-1}\int_{s_{j-1}}^{s_j}sA_j^{(i-1)}(s)\,\mathrm{d}s\end{aligned}\tag{39}$$

where $D^{(x)}(\mathbf{u})=\mathcal{DFT}\left\{d^{(x)}(\mathbf{x})\right\}$. The remaining steps for $\mathcal{DFT}\left\{\partial k(\mathbf{x};\mathbf{Q})/\partial p_i^{(x)}\right\}$ are straightforward, using the following equations in conjunction with (38).

$$\begin{aligned}\int_{s_j}^{s_{j+1}}&sA_j^{(i)}(s)\,\mathrm{d}s\\ &=\frac{s_{j+1}-s_j}{12}\Big\{\left(s_j+s_{j+1}\right)(a_1b_2+a_2b_1)\\ &\quad+s_j(3a_1b_1+a_2b_2)+s_{j+1}(a_1b_1+3a_2b_2)\Big\}\end{aligned}$$

In Fig. 14(d) and Fig. 14(f), $\partial K(\mathbf{u};\mathbf{Q})/\partial p_i^{(x)}$ and $\partial K(\mathbf{u};\mathbf{Q})/\partial p_i^{(y)}$ for $i=1$ generated using (39) are shown.

REFERENCES

[1] G. Boracchi and A. Foi, "Modeling the performance of image restoration from motion blur," *IEEE Trans. Imag. Process.*, vol. 21, no. 8, pp. 3502–3517, Aug. 2012.

[2] L. Bar, N. Kiryati, and N. Sochen, "Image deblurring in the presence of impulsive noise," *Int. J. Comput. Vis.*, vol. 70, no. 3, pp. 279–298, Dec. 2006.

[3] J.-F. Cai, R. Chan, and M. Nikolova, "Fast two-phase image deblurring under impulse noise," *J. Math. Imag. Vis.*, vol. 36, no. 1, pp. 46–53, 2010.

[4] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman, "Removing camera shake from a single photograph," in *Proc. ACM SIGGRAPH*, 2006, pp. 787–794.

[5] 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, 2008.

[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, pp. 1–10, 2007.

[7] S. Zhuo, G. Dong, and T. Sim, "Robust flash deblurring," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Dec. 2010, pp. 2440–2447.

[8] S. Cho and S. Lee, "Fast motion deblurring," *ACM Trans. Graph.*, vol. 28, no. 5, pp. 1–8, 2009.

[9] S. D. Babacan, W. Jingnan, R. Molina, and A. K. Katsaggelos, "Bayesian blind deconvolution from differently exposed image pairs," *IEEE Trans. Imag. Process.*, vol. 19, no. 11, pp. 2874–2888, Nov. 2010.

[10] J. Chen, L. Yuan, C.-K. Tang, and L. Quan, "Robust dual motion deblurring," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[11] A. Rav-Acha and S. Peleg, "Two motion-blurred images are better than one," *Pattern Regonit. Lett.*, vol. 26, no. 3, pp. 311–317, 2005.

[12] H. Asai, Y. Oyamada, J. Pilet, and S. Saito, "Cepstral analysis based blind deconvolution for motion blur," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2010, pp. 1153–1156.

[13] M. Tico and M. Vehvilainen, "Estimation of motion blur point spread function from differently exposed image frames," in *Proc. EURASIPCO*, Mar. 2006, pp. 4–8.

[14] Y. Yitzhaky and N. S. Kopeika, "Identification of blur parameters from motion-blurred images," in *Proc. Graph. Models Image Process.*, vol. 59. 1997, pp. 310–320.

[15] J. Hui and L. Chaoqiang, "Motion blur identification from image gradients," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[16] K. Patanukhom and A. Nishihara, "Identification of piecewise linear uniform motion blur," in *Proc. IEEE Region 10 Conf.*, Nov. 2007, pp. 1–4.

[17] B. Kang, J. W. Shin, and P. Park, "Piecewise linear motion blur identification using morphological filtering in frequency domain," in *Proc. ICCAS-SICE*, 2009, pp. 1928–1930.

[18] M. Ben-Ezra and S. K. Nayar, "Motion-based motion deblurring," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 6, pp. 689–698, Jun. 2004.

[19] O. Whyte, J. Sivic, and A. Zisserman, "Deblurring shaken and partially saturated images," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Dec. 2011, pp. 745–752.

[20] A. Gupta, N. Joshi, C. L. Zitnick, M. Cohen, and B. Curless, "Single image deblurring using motion density functions," in *Proc. Euro. Conf. Comput. Vis.*, 2010, pp. 171–184.

[21] M. Hirsch, C. J. Schuler, S. Harmeling, and B. Schölkopf, "Fast removal of non-uniform camera shake," in *Proc. IEEE Int. Conf. Comput. Vis.*, Feb. 2011, pp. 463–470.

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

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

[24] L. Piegl and W. Tiller, *The NURBS Book*, 2nd ed. New York, NY, USA: Springer-Verlag, 1997.

[25] H. Delingette and J. Montagnat, "Shape and topology constraints on parametric active contours," *Comput. Vis. Imag. Understand.*, vol. 83, no. 2, pp. 140–171, 2001.

[26] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int. J. Comput. Vis.*, vol. 1, no. 4, pp. 321–331, 1988.

[27] S. Osher and R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces* (ser. Applied Mathematical Sciences). New York, NY, USA: Springer-Verlag, 2003.

[28] J. A. Sethian, *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science* (ser. Cambridge monographs on applied and computational mathematics), 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 1999.

[29] R. Fletcher, *Practical Methods of Optimization*, 2nd ed. New York, NY, USA: Wiley, 1987.

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

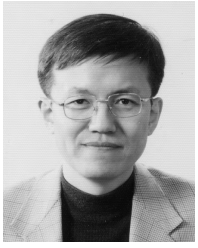
[31] L. Lucy, "An iterative technique for rectification of observed distribution," *Astronomical J.*, vol. 79, pp. 745–754, Jun. 1974.

[32] A. Levin and Y. Weiss, "User assisted separation of reflections from a single image using a sparsity prior," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 9, pp. 1647–1654, Sep. 2007.

[33] C. V. Stewart, "Robust parameter estimation in computer vision," *SIAM Rev.*, vol. 41, no. 3, pp. 513–537, 1999.

[34] L. Xu and J. Jia, "Two-phase kernel estimation for robust motion deblurring," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 150–170.

Sungchan Oh received the B.S. and M.S. degrees in electronic engineering from Sogang University, Korea, in 2006 and 2008, respectively, where he is currently pursuing the Ph.D. degree. His research interests include computer vision, pattern recognition, and real-time implementation of the computer vision algorithms.



Gyeonghwan Kim (M'95) received the B.S. (*cum laude*) and M.S. degrees in electronic engineering from Sogang University, Korea, in 1984 and 1986, respectively, and the Ph.D. degree in electrical and computer engineering from the State University of New York at Buffalo in 1996. From 1986 to 1991, he was a Researcher with the LG Precision Technology Institute, Korea. From 1993 to 1997, he was a Research Scientist with the CEDAR at SUNY, Buffalo, NY, USA. He is currently with the Department of Electronic Engineering, Sogang University, as a Professor. His research interests include pattern recognition, computer vision, hardware implementation of image processing algorithms, and man-machine interfaces.