Kernel learning for blind image recovery from motion blur

Fuqiang Qin, Shuai Fang, Lifang Wang, Xiaohui Yuan, Mohamed Elhoseny & Xiaojing Yuan

Springer

Springer

Kernel learning for blind image recovery from motion blur

Fuqiang Qin[1] · Shuai Fang[2] · Lifang Wang[1] · Xiaohui Yuan[3] · Mohamed Elhoseny[4] · Xiaojing Yuan[5]

Abstract

Restoring image from motion deblur faces great challenges in the estimation of the motion blur kernel that is the key to recover the latent sharp image. In this paper, we present a method to iteratively estimate the structural image and account for the textural component. A scale-aware smoothing operation is developed to remove fine-scale edges with resampling. Our method leverages L_0-norm regularization to enforce the sparsity of the motion blur kernel in both intensity and derivative domains. Experiments are conducted to evaluate the performance of our proposed method using two widely accepted public datasets. We found that our proposed method is insensitive to most hyper-parameters. Both qualitative evaluation and quantitative evaluation confirms that our method effectively restores the sharp image without introducing artifacts. The minimum improvements in terms of average PSNR for both datasets are more than 3.13% for all cases and the improvements in terms of average error rate are 15%. By visually comparing the estimated motion blur kernels, it is clear that the estimated kernel by our method is the closest to the actual kernel used to generate the synthesized blurry images.

1 Introduction

Restoring images from motion deblur has been a focus of research in computer vision and image processing and has a wide range of applications [15, 21, 23]. The goal is to estimate

✉ Xiaohui Yuan
xiaohui.yuan@unt.edu

1 Northwestern Polytechnical University, Xi'an, China

2 Hefei University of Technology, Hefei, China

3 University of North Texas, 3940 N. Elm, Denton, TX 76207 USA

4 Mansoura University, Mansoura, Egypt

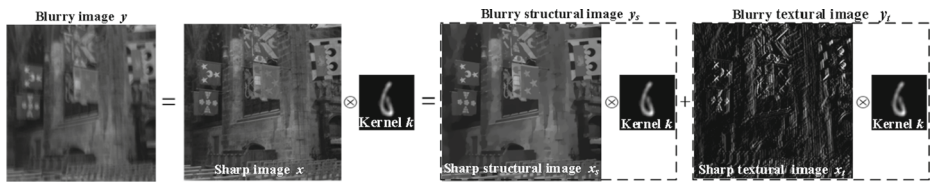5 University of Houston, Houston, TX, USA

Fig. 1 Decomposition of a blurred image

a blur kernel and recover the latent sharp image. A blurry image, denoted with y, can be represented in a form of motion-blurred sharp image x and additive noise n as follows:

$$y = x \otimes k + n \tag{1}$$

where k is the motion blur kernel and $\otimes$ denotes the convolution operation.

Studies have shown that structural edges in images are important to the estimation of blur kernels [10, 18]. Xu et al. [20] concluded that if the scale of an object is smaller than that of the blur kernel, the edge information could negatively impact the kernel estimation. Many deblurring methods [9, 10] improve blur kernel estimation by exploring the structural edges. Sharp structural edge restoration can be achieved either implicitly or explicitly. The implicit structural edge restoration methods generally employ a statistical prior as a constraint to regularize the edge selection [4, 7, 11, 12, 20, 22]. Krishnan et al. [7] proposed a normalized sparsity prior to restore images from motion blur, which tends to remove detrimental edges. Xu et al. [20] proposed a method that preserves the structural edges in the image by leveraging a L_0 regularization strategy.

The explicit structural edge restoration methods rely on the heuristic operations to recover structural edges for kernel estimation [1, 18]. In [1], bilateral filter and shock filter are performed to select structural edges, which tends to increase the computational complexity. However, such salient edges could mislead the kernel estimation process. To circumvent this issue, a two-step kernel estimation method was proposed to assess the usefulness of image edges [18].

In the aforementioned methods, structural edges are used implicitly or explicitly to estimate the motion blur kernel. Figure 1 illustrates an image that is decomposed into a structural image and a textural image. The edges in the structural image are used for kernel estimation, which uses an objective function in the following form: $\nabla x_s \otimes k - \nabla y$. This objective function computes the difference between the structural component of the blurred image and the observed image, but it neglects the fine textual components (i.e., y_t in Fig. 1).

Another issue is that the existence of fine-scale edges.[1] The scale of such edges is smaller than that of the kernel and poses a challenge to kernel estimation (a detailed explanation is in Section 2). Xu and Jia [18] proposed an edge selection scheme to reduce the suppression of fine details. The selection scheme tends to remove edges with a small gradient using the sum of the gradient magnitudes in local windows as the decision criterion. Edges with high-magnitude gradients are preserved. The magnitude of fine-scale edges tends to suffer from a significant reduction. If the image structural magnitude greatly changes after the blur, the corresponding edge information could mislead the kernel estimation.

To address the aforementioned problems, we propose a method to estimate the structural image. The structural image is used in the objective function. To circumvent the second problem, a scale-aware smoothing scheme is developed, which removes fine-scale edges with resampling. The blur kernels are usually sparse in the intensity and derivatives

---

[1]The fine-scale edge refers to edges of a short length.

<div align="center">(a)   (b)   (c)</div>
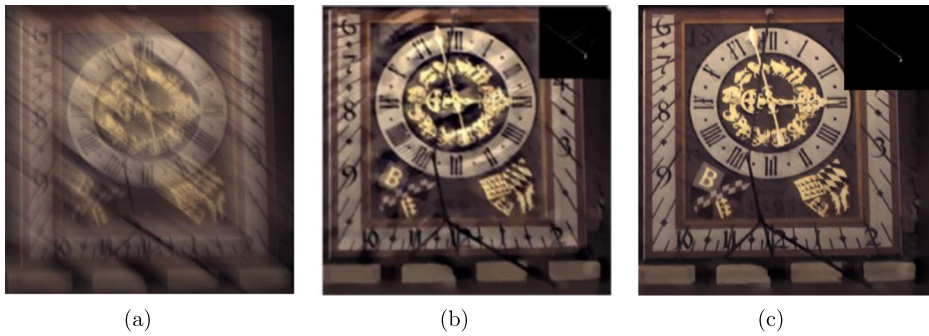
Fig. 2 Examples of the deblurred image using the derivatives of the blurred image and the derivatives of the structural component. **a** blurred image. **b** deblurred result [18] using the derivatives of the blurred image. **c** deblurred result using the derivatives of the structural component

domains. To enforce the sparsity, L_0-norm regularization is used in our method. Figure 2 illustrates an example of the motion blur image and the restored images using the method by Xu and Jia [18] (Fig. 2b) and our proposed method (Fig. 2c). The estimated motion blur kernels are superimposed on the top right corner of the result images.

The rest of this article is organized as follows. Section 2 presents our proposed method with a detailed discussion on structural image estimation and motion blur kernel generation. Section 3 discusses our experimental results using two popular image sets, which includes an ablation evaluation of the key parameters and a comparison with state-of-the-art methods. Section 4 concludes this paper with a summary.

2 Method

Our algorithm estimates the convolution kernel in a coarse-to-fine manner and refines the sharp image iteratively. Figure 3 illustrates the flowchart of our proposed method. Our
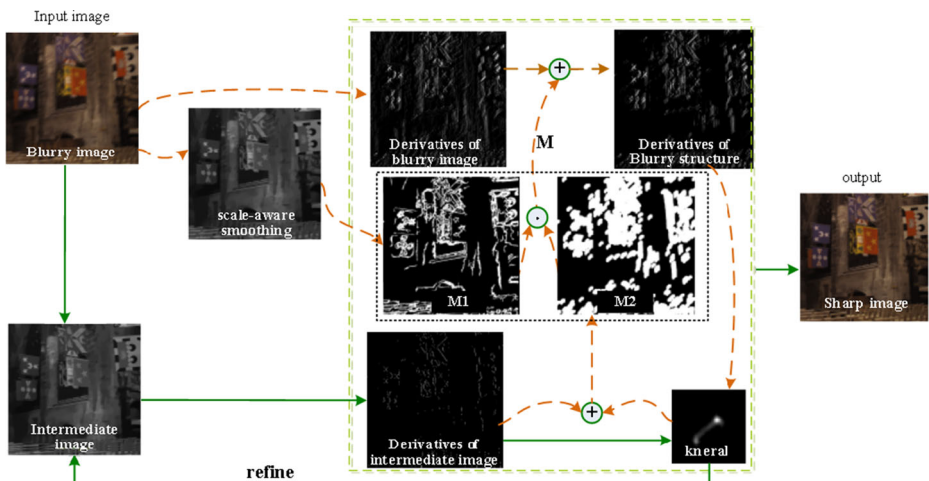

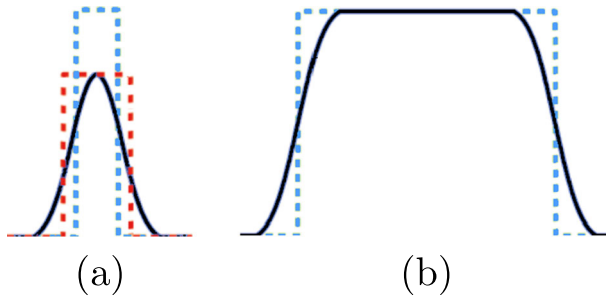**Fig. 3** The flowchart of our proposed method

Fig. 4 An illustration (recreated based on [18]) of the effect of structural edges to the estimation of kernels

method derives a sharp structural image (depicted as an intermediate image on the left), which is used for kernel estimation. Using the estimated kernel, the deblurred result is achieved by deconvolution.

Following the discussion in [18], the importance of structural image to kernel estimation is illustrated using one-dimensional step signals as shown in Fig. 4. The solid curves in Fig. 4a and b depict step signals in different scales (width). Using the same Gaussian kernel, the motion blurred results are shown with dashed curves in light blue. Because the scale of the left step signal is smaller than that of the Gaussian kernel, the signal magnitude is significantly reduced, which induces ambiguity in signal restoration. The existing methods recover a step signal with a lower magnitude depicted with a dashed curve in red. Hence, the magnitude of image structure significantly reduced by motion blur lead to imprecise kernel estimation. In comparison, the larger-scale step signal in Fig. 4b results in a precise kernel estimation.

2.1 Structural image estimation

Because the fine-scale structures (smaller than the scale of the motion blur kernel) could mislead the kernel estimation, the sum of the gradient magnitudes in a local window is used as a criterion to reduce the fine-scale edges. These schemes that only rely on gradient remove edges with a fine-scale and low-magnitude gradient as well as edges with coarse-scales and low-magnitude gradient; whereas edges with fine-scale and high-magnitude gradient tend to be preserved.

Our method uses a resampling strategy to remove fine-scale edges, which includes a down-sampling process followed by an up-sampling process. The edges with scales smaller than the down-sampling factor are removed, while the coarse-scale (long) edges are recovered by the up-sampling process. This operation is equivalent to performing a morphological opening operation in the scale space. That is, the down-sampling performs an erosion operation to eliminate the small-scale object and the up-sampling performs a dilation operation to recover the large-scale object. Figure 5 illustrates an example of this resampling process. The fine-scale structures are shown as rectangles in light gray. By down-sampling, the fine details are eliminated. The image in the middle shows the result of down-sampling. An up-sampling process recovers the image without fine-scaled structures, as shown on the right panel of Fig. 5.

The structural image contains coarse-scale edge features. In our method, we use a binary mask to pin-point pixels that correspond to structural features. In such a mask, pixels corresponding to coarse-scale structural edges are 1s. This binary mask (denoted with
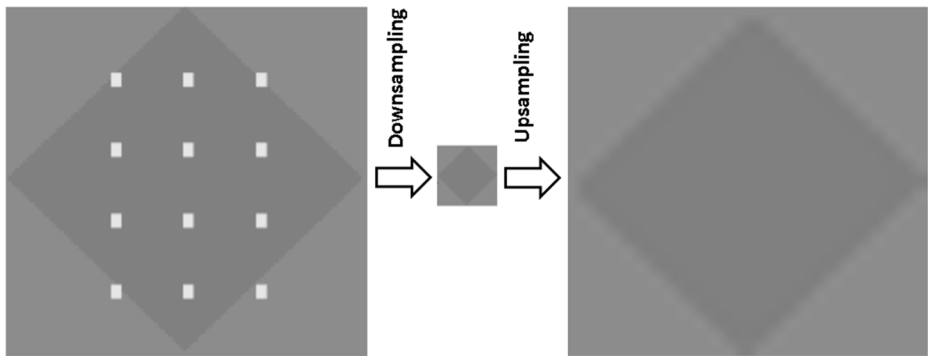
Fig. 5 Resampling operation suppresses fine-scale structures

M) consists of two components that are integrated using component-wise multiplication: $M = M_1 \odot M_2$, where $\odot$ denotes component-wise multiplication.

Binary mask M_1: M_1 aims at preserving the coarse-scale structural edges. The structural similarity (SSIM) of the resampled image and the input image is computed to distinguish the fine- and large- scale structures from the input image. We have the following three cases: (1) SSIM is very low (close to 0) at the fine-scale texture regions. This is because the small-scale textures are removed or suppressed in the resampled image. (2) SSIM is close to 1 because there is little changes in the area. (3) SSIM is in-between in the boundaries of large objects, which expresses the similarity between the clear boundary and the corresponding blurred boundary. M_1 is obtained by thresholding the SSIM map. Smooth regions and coarse-scale edges are set to 1; the rest pixels receive a value of 0. The threshold is proportional to the scaling factor of the down-sampling operator. In this paper, the threshold is 0.1 with the scaling factor at 0.25.

Binary mask M_2: In the derivatives domain, binary mask M_2 is obtained by thresholding the convolution result of the intermediate image with the corresponding Gaussian kernel. The element are 1 for the regions corresponding to large-gradient edges and 0 for the rest. The intermediate sharp image is the result of L_0 regularization, in which edges with small gradients are removed. In the smooth regions of an image, the pixel value varies slightly. Our choice of the threshold considers the slight differences of pixel values and the threshold for M_2 is 0.001.

We divide an image into textural (representing the fine-scale components) and structural (representing the coarse-scale components) components. The derivatives of the blurry image are computed as a summation of the gradient of structural component y_s and the gradient of textural component y_t:

$$\sum_{d \in \{h, v\}} (f_d \otimes y_s + f_d \otimes y_t) = \sum_{d \in \{h, v\}} (f_d \otimes x_s \otimes k + f_d \otimes x_t \otimes k) \qquad (2)$$

where f_d consists of a horizontal gradient operator f_h and a vertical gradient operator f_v, x_s and x_t are the structural and textural components of the clear image. The blur kernel is estimated using the coarse-scale structural component ∇y_s:

$$\nabla y_s = \sum_{d \in \{h, v\}} f_d \otimes x_s \otimes k \qquad (3)$$

Following the L_0 regularization, textural components in the intermediate sharp image are greatly suppressed. Therefore, image x is a plausible approximation of the structural image x_s. Accordingly, y_s and ∇y_s are approximated with $M \odot y$ and $M \odot (f_d \otimes y)$, respectively.

2.2 Kernel estimation

Provided with the motion blur kernel k, we minimize the following function to estimate the clear image x:

$$x = \arg \min_x ||x \otimes k - y||_2^2 + \alpha ||f_h \otimes x + f_v \otimes x||_0 \tag{4}$$

where f_h is the first derivative in the horizontal direction of the image and f_v is the first derivative of in the vertical direction of the image, α is a scalar weight factor and function $|| \cdot ||_0$ computes the number of non-zero values.

The kernel k is estimated by minimizing the following objective function:

$$k = \arg \min_k ||\nabla x \otimes k - M \odot \nabla y||_2^2 + \lambda (\sigma ||k||_0 + ||C(k)||_0) \tag{5}$$

where

$$\nabla x = (f_h \otimes x, f_v \otimes x)^T,$$

$$\nabla y = (f_h \otimes y, f_v \otimes y)^T,$$

$$C(k) = |f_h \otimes k| + |f_v \otimes k|,$$

and $\odot$ denotes component-wise multiplication. M is a binary mask, in which 0 and 1 represent the texture and structure components, respectively.

To minimize (5), our method follows the half-quadratic splitting method [10] and the objective function is rewritten as follows:

$$k = \arg \min_{k,u,g} ||\nabla x \otimes k - M \odot \nabla y||_2^2 + \beta ||k - u||_2^2 + \mu ||\nabla k - g||_2^2 + \lambda (\sigma ||u||_0 + ||C(g)||_0) \tag{6}$$

where β and μ are self-adapt in the optimization process and the values vary in the range of [4 256] (refer to [10, 16, 19] for a detailed explanation of the self-adapting process). The process consists of the following two iterative steps:

Step 1:   Calculate k while fixing u and g. The objective function is rewritten as follows:

$$||\nabla x \otimes k - M \odot \nabla y||_2^2 + \beta ||k - u||_2^2 + \mu ||\nabla k - g||_2^2 \tag{7}$$

Since this objective function is quadratic, we derive the closed-form solution using Fourier transform as follows:

$$k = \mathcal{F}^{-1} \left(\frac{\sum_{d \in \{h,v\}} (\mathcal{F}_{x \times y} + \mu \mathcal{F}_{f \times g}) + \beta \mathcal{F}(u)}{\sum_{d \in \{h,v\}} (\mathcal{F}_{x \times y} + \mu \mathcal{F}_{f \times f}) + \beta} \right) \tag{8}$$

where

$$\mathcal{F}_{x \times y} = \mathcal{F}^H (f_d \otimes x) \mathcal{F}(M \odot (f_d \otimes y)),$$

$$\mathcal{F}_{x \times x} = \mathcal{F}^H (f_d \otimes x) \mathcal{F}(f_d \otimes y),$$

$$\mathcal{F}_{f \times g} = \mathcal{F}^H (f_d) \mathcal{F}(g_d),$$

$$\mathcal{F}_{f \times f} = \mathcal{F}^H (f_d) \mathcal{F}(f_d),$$

$\mathcal{F}$ denotes Fourier transform and $\mathcal{F}^H$ denotes the conjugate Fourier transform.

Step 2:   Calculate u and g while fixing k.

$$u = \begin{cases} k, & |k|^2 \geq \frac{\lambda\sigma}{\beta} \\ 0, & \text{Otherwise} \end{cases} \tag{9}$$

$$g = \begin{cases} \nabla k, & |\nabla k|^2 \geq \frac{\lambda}{\mu} \\ 0, & \text{Otherwise} \end{cases} \tag{10}$$

3 Experimental results and discussion

3.1 Datasets and settings

To evaluate the performance of our proposed method, we adopt the two sets of images and the motion blur kernels: image set 1 by Sun et al. [14] and image set 2 by Köhler et al. [6]. The image and motion blur kernels can be accessed at http://cs.brown.edu/~lbsun/deblur2013/deblur2013iccp.html and http://webdav.is.mpg.de/pixel/benchmark4camerashake/. In our experiments, the scaling factor is 0.25 for the scale-aware filtering. The threshold to generate the binary masks M_1 and M_2 are 0.1 and 0.001, respectively.

3.2 Ablation experiment

Our objective function includes three regularization terms: L_0 regularization term α, kernel sparse regularization term, and kernel gradient sparse regularization term. In this section, we evaluate each by varying the value of one term while fixing the others. In the evaluation of α, we change the value of α in the range of 0 to 0.002 while keeping the other terms constant. Figure 6 shows the deblurred results and the corresponding PSNR is reported in Table 1. In general, as α increases PSNR reduces, and our method achieves the best PSNR at $\alpha = 10^{-4}$. In our experiments, however, the value of α was not fixed. At the beginning of the iteration, we used a large α to smooth the image and suppress the errors and ringing artifacts. As the optimization process continues, the blurry kernel gets more accurate, which requires a smaller α to preserve structural information. Hence, the value for α decreases gradually. In our experiments, α decreases from 10^{-3} to 10^{-4}.

For the kernel sparse regularization term $\lambda\sigma$, it changes in the range of 10^{-5} and 10^{-2}. Following the same scheme, we got the following PSNRs for various $\lambda\sigma$. Table 1 list the average PSNR for various $\lambda\sigma$ values. The difference is small and at $\lambda\sigma = 10^{-4}$ the best PSNR was achieved. Hence, in the rest of our experiments, $\lambda\sigma$ was set to 10^{-4}.

For kernel gradient sparse regularization term λ, we change its value in the range of [0.1 50] and the results are in the following table. The difference between PSNRs is small. Based on these results, the choice for λ is 1.

To understand the progressive improvement of our method, we visualize the intermediate structural image and the motion blur kernel of some iterations in Fig. 7. From left to right, the panels show the intermediate results as the iteration number increases. At an early stage (left panels), the intermediate images contains strong artifacts. As the optimization continues, such artifacts are greatly suppressed and the structural information is recovered (right panels). The progressive improvement is also evident in the estimated motion blur kernel as shown on the top left corner of each image.
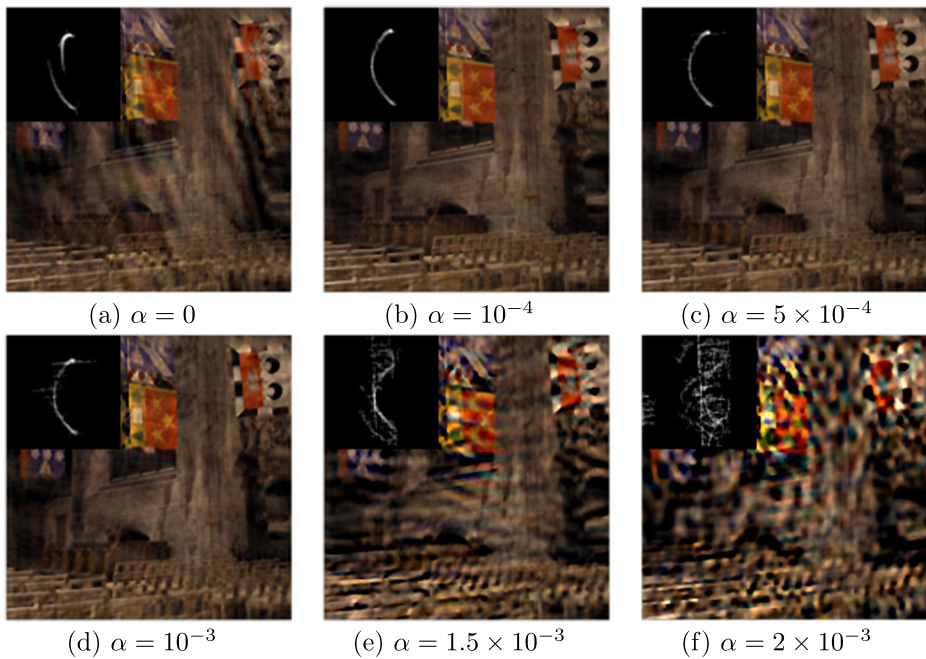
Fig. 6 The deblurred results using different α values

3.3 Qualitative evaluation

Figure 8 presents a visual comparison of the deblurred results of the Clock image in dataset 2 [6] using five state-of-the-art methods and our proposed method. Figure 8a and e show the sharp image and the synthesized motion blurred image. The zoom-in views of regions enclosed by two boxes are shown at the bottom of each panel. The one on top left corner depicts a high-contrast region; the one on the right margin shows a low-contrast feature. Except for Fig. 8b, the restored images by the other methods appear to be clear and sharp. However, when we look into the details of the deblurred results, the blurry marks are noticeable in the zoom-in views of Fig. 8c, d, f, and g: the two bright stripes near number six and the fuzzy appearance of number 52. The proposed method exhibits a much superior result, as shown in Fig. 8h. In comparison with state-of-the-art methods, it is evident that our algorithm performs well on the recovery of details while suppressing ringing artifacts.

Table 1 PSNR of regularization terms

α	0	10^{-4}	5×10^{-4}	10^{-3}	1.5×10^{-3}	2×10^{-3}
PSNR	23.83	27.21	26.61	26.15	20.19	18.57
$\lambda\sigma$	0	10^{-5}	10^{-4}	10^{-3}	5×10^{-3}	
PSNR	27.41	28.05	28.22	28.03	27.45	
λ	0	10^{-5}	10^{-4}	10^{-3}	5×10^{-3}	
PSNR	28.12	28.33	28.58	28.42	28.25	

Fig. 7 The structural image and the motion blur kernel (top-left of each panel) of each iteration

Figure 9 depicts the deblurred result using five state-of-the-art methods and ours using an image from dataset 1 [14]. Figure 9a and (e) illustrate the sharp image and the synthesized motion-blur image. The kernel used to generate Fig. 9e is shown on the top right corner of Fig. 9a. Zoom-in views of two regions (marked with a red and a blue box) are depicted at the bottom of each panel. It is clear that the results are shown in Fig. 9b and f suffer from severe aliasing effects. The results of Levin et al. [9] and Sun et al. [14] (as shown in Fig. 9c and d) achieve better results. Although the spatial details still show ringing artifacts (see the zoom-in views), such artifacts are mitigated in Fig. 9c, d and g. In contrast, the result of our proposed method demonstrates a much clear reconstruction with great details and fewer artifacts. By visually comparing the estimated motion blur kernels shown on the top left corner of each panel, it is clear that the estimation produced by our proposed method is the closest to the actual kernel used to generate the synthesized blurry images shown in Fig. 9e.

3.4 Quantitative evaluation

Our quantitative evaluation adopts the Peak Signal Noise Ratio (PSNR) and Error Ratio (Er):

$$PSNR(x, x_d) = 10 \log_{10}(\max(x)^2 / MSE(x, x_d)), \tag{11}$$

$$Er(x, x_g, x_d) = \frac{MSE(x_g, x)}{MSE(x_d, x)}, \tag{12}$$

where x_d and x_g are deblurred image and the ground-truth image, respectively. MSE computes the mean square error: $MSE(a, b) = \frac{1}{MN} \sum_{i,j} (a(i, j) - b(i, j))^2$, where M and N
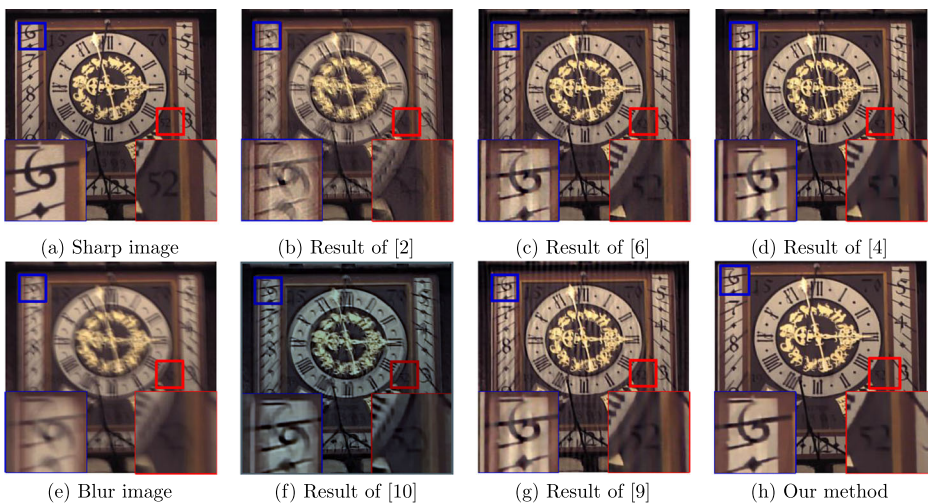

| (a) Sharp image | (b) Result of [2] | (c) Result of [6] | (d) Result of [4] |
| (e) Blur image | (f) Result of [10] | (g) Result of [9] | (h) Our method |

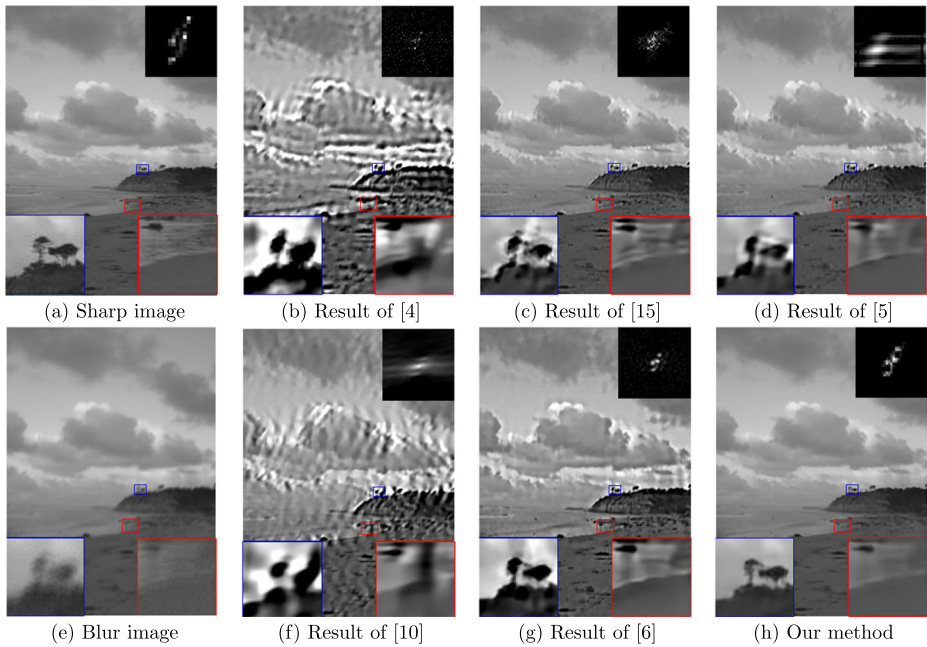**Fig. 8** Deblurred results of an exemplar image in dataset 2 [6]

Fig. 9 Deblurred results of an exemplary image in dataset 1 [14]

are the number of rows and columns of the image, i and j are the pixel indices. The minimum error ration is one and a large error ratio implies poor image quality.

Using the 80 sharp images in dataset 1 [14], we synthesized 640 motion-blurred images using eight motion blur kernels. In addition, Gaussian noise is introduced into the blurred image and the variance of the Gaussian function is set to 0.01. The results of PSNRs and error rates are reported in Table 2. Using the four sharp images in dataset 2 [6], 48 blurry images were synthesized using 12 blur kernels. The results are reported in Table 3.

Table 2 reports the average PSNRs and error rates of the restored image from the 640 cases. The first column presents the PSNR and error ratio of the deblurred results using the motion blur kernel that was used to generate the blurry images. This is the best possible result achievable if the exact motion blur kernel is recovered. By comparing the results of all the methods, it is clear that our method yielded the greatest PSNR (31.53) and the smallest error rate (1.69), which is closest to the values in the first column. Compared to the second-best PSNR produced by [8], our method achieved an improvement of 6.5%. In contrast to the second-best error ratio produced by [14], our method exhibited an improvement of 15%. It is evident that the kernels estimated by our proposed method are much more accurate than the state-of-the-art methods.

Table 2 Average PSNR and error ratio of our proposed method and the state-of-the-art methods using dataset 1 [14]. Ref. is the deblurred result using the actual motion blur kernel

	Ref.	[1]	[2]	[7]	[9]	[18]	[14]	[8]	Ours
PSNR	32.46	26.24	20.17	23.22	24.94	28.31	29.56	29.61	31.53
Er	1.00	4.19	18.14	8.37	5.65	2.60	1.95	2.07	1.69

Table 3 Average PSNR of the restored images of our proposed method and the state-of-the-art methods using dataset 2 [6]

Images	[17]	[5]	[13]	[3]	[7]	[1]	[18]	[24]	Ours
Church	27.55	26.72	26.43	22.78	26.87	28.91	29.41	30.13	**31.73**
Clock	22.87	22.59	20.60	14.94	21.76	24.27	**25.48**	25.47	25.22
Backyard	28.61	26.42	25.88	22.97	26.64	29.20	29.30	30.18	**31.46**
Roof	24.71	23.54	22.40	22.97	22.87	26.61	26.76	26.77	**27.60**
Mean	25.93	24.82	23.82	18.90	24.53	27.25	27.74	28.12	**29.00**

Table 3 presents the average PSNRs for each image in dataset 2 [6] given 12 motion blur kernels. Each row lists the average PSNRs of the synthesized blurry variants of one sharp image. We highlight the best results in bold font. Out of four cases, our method achieved the best performance in three and the maximum improvement rate compared to the second-best is 5.31%. There is one case, in which the method by Xu and Jia [18] resulted in the greatest PSNR. However, the difference to our method is within 1%. The last row of this table reports the mean of all cases. The mean PSNR of our method is 29 and the second best is 28.12 (achieved by Yue et al. [24]), which is an improvement of 3.13%.

Note that these two datasets account for a variety of motion blur. The improvements in terms of PSNR for the experiments with the two datasets are mostly consistent. It is hence evident that the proposed method effectively restores the sharp image without introducing many artifacts.

4 Conclusion

Restoring images from motion deblur has a wide range of computer vision applications. The key is estimating a motion blur kernel to recover the latent sharp image. In this paper, we present a method to estimate the structural image and use it in our objective function. A scale-aware smoothing operation is developed to remove fine-scale edges with resampling. Our method leverages L_0-norm regularization to enforce the sparsity of the motion blur kernel in both intensity and derivative domains. Experiments are conducted to evaluate the performance of our proposed method using two widely accepted public datasets.

By conducting ablation experiments, we found that our proposed method is insensitive to most hyper-parameters. In particular, the PSNR varies in a small range for different values of λ and $\lambda\sigma$ (two key parameters in our method). In addition, to accelerate the optimization speed, our method gradually reduces the value of α to ensure the suppression of artifacts and preservation of structural information.

By visualizing the results of our proposed method and the state-of-the-art, our method resulted in much clearer reconstructions with great details and fewer artifacts. By visually comparing the estimated motion blur kernels, it is clear that the estimated kernel by our method is the closest to the actual kernel used to generate the synthesized blurry images. Using PSNR and error rate, we also quantitatively compared our method with the state-of-the-art. The minimum improvements in terms of average PSNR for both datasets are more than 3.13% and the improvements in terms of average error rate are 15%. Given that the two datasets account for a variety of motion blur, our method effectively restores the sharp image without introducing artifacts.

Compliance with Ethical Standards

Conflict of interests   The authors declare no potential conflict of interest.

References

1. Cho S, Lee S (2009) Fast motion deblurring. ACM Trans Graph 28(5):145:1–145:8
2. Cho TS, Paris S, Horn BKP, Freeman WT (2011) Blur kernel estimation using the radon transform. In: IEEE conference on computer vision and pattern recognition, pp 241–248
3. Fergus R, Singh B, Hertzmann A, Roweis ST, Freeman WT (2006) Removing camera shake from a single photograph. ACM Trans Graph 25(3):787–794
4. Gong D, Tan M, Zhang Y, Hengel AVD, Shi Q (2016) Blind image deconvolution by automatic gradient activation. In: IEEE conference on computer vision and pattern recognition, pp 1827–1836
5. Hirsch M, Schuler CJ, Harmeling S, Schölkopf B (2011) Fast removal of non-uniform camera shake. In: International conference on computer vision, pp 463–470
6. Köhler R, Hirsch M, Mohler B, Schölkopf B, Harmeling S (2012) Recording and playback of camera shake: Benchmarking blind deconvolution with a real-world database. In: Proceedings of the 12th European Conference on Computer Vision - Volume Part VII, pp 27–40, Florence, Italy
7. Krishnan D, Tay T, Fergus R (2011) Blind deconvolution using a normalized sparsity measure. In: International conference on computer vision and pattern recognition, pp 233–240
8. Lai W, Ding J, Lin Y, Chuang Y (2015) Blur kernel estimation using normalized color-line priors. In: IEEE Conference on computer vision and pattern recognition, pp 64–72
9. Levin A, Weiss Y, Durand F, Freeman WT (2009) Understanding and evaluating blind deconvolution algorithms. In: 2009 IEEE conference on computer vision and pattern recognition, pp 1964–1971
10. Pan J, Hu Z, Su Z, Yang M (2014) Deblurring text images via l0-regularized intensity and gradient prior. In: IEEE conference on computer vision and pattern recognition, pp 2901–2908
11. Pan J, Hu Z, Su Z, Yang M (2017) l_0 -regularized intensity and gradient prior for deblurring text images and beyond. IEEE Trans Pattern Anal Mach Intell 39(2):342–355
12. Pan J, Sun D, Pfister H, Yang M (2018) Deblurring images via dark channel prior. IEEE Trans Pattern Anal Mach Intell 40(10):2315–2328
13. Shan Q, Jia J, Agarwala A (2008) High-quality motion deblurring from a single image. ACM Trans Graph 27(3):73:1–10
14. Sun L, Cho S, Wang J, Hays J (2013) Edge-based blur kernel estimation using patch priors. In: IEEE international conference on computational photography (ICCP), pp 1–8
15. Wang W, Yuan X, Wu X, Liu Y (2017) Fast image dehazing method based on linear transformation. IEEE Trans Multimed 19(6):1142–1155
16. Wang Y, Yang J, Yin W, Zhang Y (2008) A new alternating minimization algorithm for total variation image reconstruction. SIAM Imaging Sciences 1(3):248–272
17. Whyte O, Sivic J, Zisserman A (2011) Deblurring shaken and partially saturated images. In: IEEE international conference on computer vision workshops, pp 745–752
18. Xu L, Jia J (2010) Two-phase kernel estimation for robust motion deblurring. In: Proceedings of the 11th European conference on computer vision: Part I, pp 157–170
19. Xu L, Lu C, Xu Y, Jia J (2011) Image smoothing via L0 gradient minimization. ACM Trans Graph 30(6):174–179
20. Xu L, Zheng S, Jia J (2013) Unnatural l0 sparse representation for natural image deblurring. In: IEEE conference on computer vision and pattern recognition, pp 1107–1114
21. Yuan X (2010) Segmentation of blurry object by learning from examples. In: Proceedings of SPIE, 7623, 7623–9
22. Yuan X, Kong L, Feng D, Wei Z (2017) Automatic feature point detection and tracking of human action in time-of-flight videos. IEEE/CAA Journal of Automatica Sinica 4(4):677–685

23. Yuan X, Li D, Mohapatra D, Elhoseny M (2018) Automatic removal of complex shadows from indoor videos using transfer learning and dynamic thresholding. Comput Electr Eng 70:813–825
24. Yue T, Cho S, Wang J, Dai Q (2014) Hybrid image deblurring by fusing edge and power spectrum information. In: European conference on computer vision, vol 8695, pp 79–93

Publisher's note   Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Fuqiang Qin received a B.S. degree from the School of International Business, Qingdao University in 2007, and an M.S. degree from the School of Information, Flinders University and Nankai University (a joint program), Beijing, China in 2016. He is currently a Ph.D. student at the School of Computer Science and Engineering, Northwestern Polytechnical University, Xi'an, China. His research areas include remote sensing big data analysis, image processing based on deep learning, and GIS geographic information system development.

Shuai Fang received a Ph.D. degree in Pattern Recognition from Northeastern University, Shenyang, China, in 2005. She is currently a Professor at the School of Computer Science and Information Engineering, Hefei University of Technology, Hefei, Anhui, China. Her research interests include pattern recognition, computer vision, and image processing.

Lifang Wang received a Ph.D. degree in Computer Science and Technology from Northwestern Polytechnical University, Xi'an China. She is a Professor at the School of Computer Science, at Northwestern Polytechnical University. Her research interests include electronic commerce technology, cloud computing, cloud storage, machine learning, deep learning, and natural language processing.
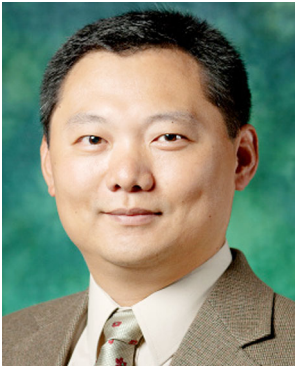
Xiaohui Yuan received a B.S. degree in Electrical Engineering from the Hefei University of Technology, Hefei, China in 1996 and a Ph.D. degree in Computer Science from Tulane University, New Orleans, the USA in 2004. He is currently an Associate Professor at the Department of Computer Science and Engineering at the University of North Texas and a Visiting Professor at the China University of Geosciences, Wuhan, China. His research interests include computer vision, data mining, machine learning, and artificial intelligence. Dr. Yuan is a recipient of the Ralph E. Powe Junior Faculty Enhancement award in 2008 and the Air Force Summer Faculty Fellowship in 2011, 2012, and 2013. He is a senior member of IEEE.

Mohamed Elhoseny is currently an Assistant Professor at the Faculty of Computers and Information, Mansoura University. He authored/co-authored many ISI Journal articles in high-ranked and prestigious journals and authored/edited more than 15 international books. He serves as the Editor-in-Chief of the International Journal of Smart Sensor Technologies and Applications (IGI Global), and is an Associate Editor of many journals such as IEEE Journal of Biomedical and Health Informatics, IEEE Access, Scientific Reports (Nature), IEEE Future Directions, Remote Sensing, and International Journal of E-services and Mobile Applications (IGI Global). He is the Editor-in-Chief of the Studies in Distributed Intelligence Springer Book Series, the Sensors Communication for Urban Intelligence, and the Distributed Sensing and Intelligent Systems CRC Press-Taylor & Francis Book Series. He received the Young Researcher Award in Artificial Intelligence in 2019, Obada International Prize for young distinguished scientists 2020, the Egypt National Prize for Young Researchers in 2018, the best Ph.D. thesis in Mansoura University in 2015, the SRGE best young researcher award in 2017, and the membership of The Egyptian Young Academy of Science in 2019.

Xiaojing Yuan is a Professor at the Department of Engineering Technology at the University of Houston. Her research interests include signal processing and wireless sensor networks.