Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/MBE

MBE, 16(5): 4036–4052.
DOI: 10.3934/mbe.2019199
Received: 25 January 2019
Accepted: 14 April 2019
Published: 07 May 2019

*Research article*

# An efficient motion deblurring based on FPSF and clustering

**Hui-Yu Huang*and Wei-Chang Tsai**

Department of Computer Science and Information Engineering, National Formosa University, Huwei, Yunlin 632, Taiwan

* **Correspondence:** Email: hyhuang@nfu.edu.tw; Tel: +88656315590; Fax:+88656330456.

**Abstract:** A blurry photograph is a type of degradation of image quality. Blur could be modeled with convolutional operation of an image with a blurring kernel, also known as the point spread function (or PSF). Image deconvolution is the process of recovering the unknown image from its blurred version, given a blurring kernel. It is quite time-consuming by using the recursive process to estimate the kernel. The work proposes an approach to the blurred image into a sharp image using the intelligent computing integrating with linear image degradation process, Fourier transforms, and Fourier spectrum. Based on the model built from the information of the blurry image, a linear degradation process, we estimate the kernel power spectrum and compute phase-retrieval applied the intelligent computing for Fourier theorem and Wiener-Khinchin theorem. Then, the optimal blur kernel can be estimated by using kernel clustering and kernel integration under fast point spread function (FPSF). Finally, the sharp image is achieved by using a deconvolution process, inverse Fourier transform. The approach to deblurring is applied the intelligent computing on the estimated image and Peak signal-to-noise ratio (PSNR) to evaluate the performance. By rebuild an improved PSF, the computing strategy leads a kernel estimation of the caught image and reduces the computational time. Experimental results demonstrate that the proposed method with intelligent computing applied can decrease computational time and achieve good visual quality for deblurring images.

**Keywords:** image deblurring; blur kernel; point spread function(PSF); image restoration; phase retrieval

## 1. Introduction

In an information era, owing to hardware and software advances, the development of technologies in engineering is tented to high computational ability. Intelligent computing and machine learning provide a good strategy to solve the complicated problems and a fast computational ability to reduce the complexity of the procedure. In the computer vision field, it usually operates in the frequency domain to extract the essential information in order to model the events. However, the procedure operated in the frequency domain is a kind of complex computation. Involving the intelligence into the modeling technique to solve the confronted problems of computer vision is the current trend. Intelligent computing is the integration of many technologies and decisions of knowledge for the computing environment. Computational photography scheme, image processing, requests the intelligent computing to deal with the object model and feature extraction. Motion blur is usually presented in many practical scenarios, such as hand-held cameras especially mobile material. The motion-blurred picture is caused by the relative motion between the camera and an imaged scene during exposure. Generally, the relative motion can be divided into two kinds: camera shake and object motion. Sensor movement during exposure leads to unwanted blur in the acquired image. Assuming a scene and ignoring the effects of defocus and lens abnormality, each point in the blurred image can be modeled as the convolution of the unblurred image by a global point spread function (PSF). Image deblurring aims to achieve a deconvolution process to recover the clear image from the acquired blurry image. Based on intelligent computing strategy, image deblurring could be reduced the computing complexity and improved the computational time.

Artificial neural networks (NNs) have been used extensively in image processing [1, 2]. Schuler *et al.* [3] proposed a learning-based neural network (NN) to estimate the features and then estimate blur kernel for deblurring by using the deconvolution. Additionally, a common assumption in motion deblurring methods is that the motion PSF is spatially invariant. This implies that all pixels are convolved with the same motion blur kernel. The problem of blur kernel estimation and more generally blind deconvolution is a long-existing problem in computer vision. Restoration of blurry images is highly dependent on estimation of motion blur kernel after implementing the appropriate image restoration method. Many well-known PSF algorithms to estimate blur kernel have been proposed [4, 5, 6, 7, 8]. When the PSF is known, or can be estimated, a deconvolution algorithm, such as Richardson-Lucy [9, 10, 11, 12], can be used to deblur the image. Motion blur is a type of the relative motion of the camera and shooting scene during exposure. Mathematically, the corresponding motion blur information is usually modeled as a linear image degradation process.

$$B = I \otimes K + N, \tag{1.1}$$

where $\otimes$ denotes the convolution operator, $B$, $I$, $K$, and $N$ denote the blurred image, true sharp image, unknown blur kernel, and noise term, respectively. Blind image deconvolution is an inherently ill-posed problem since the blurred image $B$ does not provide enough information for determining both $I$ and $K$. Therefore, how to estimate blur kernel $K$ from blurred image $B$ is an important issue in motion-blurred image restoration. Many of studies have been proposed. Some studies are briefly described in the following.

Blurred kernel information is usually hidden in the regions with edges, if the edge of an image suffered serious damage, it will result in inaccurate blur kernel estimation. Tai *et al.* [10] proposed a modified Richardson-Lucy (RL) method to incorporate space-invariant blur model under a projective motion path for image scenes. Yang et al. [11] proposed blur kernel estimation and non-blind image deconvolution to deblur image by using bilateral filter and Gradient attenuation Richardson-Lucy deconvolution algorithm. Dobeš *et al.* [13] and Goldstein and Fattal [5] proposed the kernel estimation in frequency domain. The blur kernel is then recovered using a phase retrieval algorithm with improved convergence and disambiguation capabilities. Deblurring approaches, based on spectral properties and edge information of an image have presented by [5, 14, 15] to retrieval the blur kernel information. In addition, one of many deblurring techniques is to incorporate image priors to impose on the deblurred results. Deshpande and Patnaik [16] proposed an image motion deblurring algorithm based on dual Fourier spectrum combined with bit plane slicing algorithm and Radon transform (RT) for accurate estimation of PSF parameters (blur length and blur angle). Shao *et al.* [17] used non-stationary Gaussian prior to estimate the salient edges of image as the cues to blur kernel estimation. He *et al.* [18] proposed motion blurring which used different priors for the local region and the motion blur kernel to formulate a minimization energy function that alternates between blur kernel estimation and deblurring image restoration. Jia [19] relied on color mixtures to estimate the motion blur kernel of moving objects given their boundary alpha values. Levin *et al.* [20] used a maximum a posteriori estimation (MAP) to estimate blur kernel and achieve the deblurring results. Except MAP methods [21, 22], many methods are being developed [13, 23, 24, 25, 26].

In this paper, we propose a motion deblurring method based on fast PSF (FPSF) to achieve image restoration. The advantages of this system can speed up the running time and find an optimal blur kernel, as well as obtain a good image quality for deblurring. In addition, in order to verify the reliability of our proposed system, the experimental data include the real motion-blurred images and artificial blurred images.

The rest of this paper is organized as follows. Section 2 describes the related techniques. Section 3 describes the proposed method which includes blur kernel clustering, blur kernel integration, and the optimal blue kernel searching. Section 4 presents the experimental results. Finally, a conclusion is given in Section 5.

## 2. Related techniques

In this section, we briefly describe the techniques related to our proposed approach.

### 2.1. Blur kernel spectrum estimation

A particular property of natural image scenes can be illustrated by the following power-law relationship [27, 28].

$$| \hat{I}(\omega) |^2 \propto \| \omega \|^{-\beta}, \tag{2.1}$$

where $\omega$ is the coordinates in frequency domain, $\hat{I}$ denotes Fourier transform of a natural image ($I$), according to the literature, $\beta \approx 2$.

As the researches indicated, the blur information is hidden in the power-law of the neighborhoods of edges, therefore, filtering an image forming Eq. (2.1) can be acquired this information and is expressed as

$$|\widehat{I * d}(\omega)|^2 = |\hat{I}(\omega)|^2 \cdot |\hat{d}(\omega)|^2 \approx \pi^2 \| \omega \|^{-2} \| \omega \|^2 = c, \tag{2.2}$$

for constant $c$, $d$ is a first-order Laplacian filter. Thus, a blurry image $B = I * k$, this filtering process can be used to estimate the following blur-kernel power spectrum $|\hat{k}(\omega)|^2$.

$$|(\widehat{B * d})(\omega)|^2 = |\hat{I}(\omega)|^2 \cdot |\hat{d}(\omega)|^2 \cdot |\hat{k}(\omega)|^2 \approx c|\hat{k}(\omega)|^2. \tag{2.3}$$

The power spectrum of any signal $F$ can relay to its autocorrelation according to the Wiener-Khinchin theorem [29].

$$\hat{R}_F(\omega) = |\hat{F}(\omega)|^2, \tag{2.4}$$

where the autocorrelation is defined by $R_F(x) = (\bar{F} * F)(x)$. The blur approximation in Eq. (2.3) can be identified by real-space parts for the spectrum and can be expressed as

$$R_{B*d}(x) \approx cR_k(x). \tag{2.5}$$

Evidently, the power spectrum of a natural image varies by multiplicative factors along the different directions, that is,

$$|\hat{I}(\omega)|^2 \approx c_{\theta(\omega)} \cdot \parallel \omega \parallel^{-2}, \tag{2.6}$$

where $\theta(\omega) = \arctan(\omega_x, \omega_y)$ is the angle of the vector $\omega$.

Blur kernel information, like $c_{\theta(\omega)}$ and the kernel phase, can be recovered by means of the Fourier slice theorem and Wiener-Khinchin theorem, given autocorrelation functions computed from the input blurry image $B(x)$. From Eq. (2.6) known, only single parameter $c_\theta$ is unknown. Based on Fourier theorem and Wiener-Khinchin theorem, Eq.(2.3) can be rewritten in real-space and expressed as Eq.(2.7) [5].

$$f_\theta(x) \approx c_\theta \cdot R_{P_\theta(k)}(x), \theta \in [-\pi, \pi], \tag{2.7}$$

where $P_\theta$ is a projection of a 2D signal into 1D by integrating it along the direction orthogonal to $\theta$. By repeating this procedure for all the $\theta$, an approximation for the 2D blur-kernel power spectrum function $|\hat{k}(\omega)|^2$ can be obtained. In [5], this procedure repeated three times. In addition, based an iterative phase-retrieval algorithm, this approximation can recover the blur kernel $k$.

### 2.2. Iterative phase retrieval

The phase recovery also called phase retrieval. As described above, recovering the kernel $k$, given its power spectrum $|\hat{k}|^2$ requires estimating the phase component of $\hat{k}(\omega)$. However, this procedure only obtains the spectrum information, the phase information is still unknown because it iteratively switches between Fourier and real-space domains. In addition, the input $|\hat{k}|^2$ and the spatial constraints may not guarantee a unique solution. Moreover, as [30] discussed, it may converge to local minima formation, therefore, the phase retrieval procedure is repeated multiple times under starting randomly phase component to further estimate the blur kernel.

According to GS algorithm [31], this algorithm is a common method for phase retrieval. It is based on iterative Fourier transform and inverse Fourier transform between the object domain and the Fourier domain. A hybrid input-output method is used to estimate the blur kernel in iterative phase retrieval procedure under the appropriate frequency/spatial domain constraints [5, 31]. Therefore, based on iterative phase retrieval algorithm, the blur kernel can be recovered. Thus, the blurry image can be deblurred through a deconvolution. The procedures of phase retrieval algorithm are briefly described as follows. Its pseudo code is shown in Algorithm 1 (see [5] for details).

**Step 1)** Given the initial phase $\phi(\omega)$ within $[-\pi, \pi]$ randomly generated.

**Step 2)** Transfrom the real-space domain $g$ using inverse Fourier transform.

**Step 3)** Transform $g$ into $\hat{g}$ using Fourier domain constraints.

**Step 4)** Transform to real-space $g_2$ using the phase information of $\hat{g}$.

**Step 5)** Compute the constraint $R(x)$ using the space domain constraints.

**Step 6)** Obtain a hybrid input-output constraint $\Omega$ which is the union of $R(x)$ and $s$.

**Step 7)** Repeat Steps 3 - 6 for $m$ times, and output the $k_n$ based on the constraint $\Omega$ which is the union of $g_2(x)$ and $s$.

**Algorithm 1** Iterative phase retrieval

```
Input: kernel magnitude spectrum, $p(\omega) =| \hat{k}(\omega) |$ and kernel size = $s$
for $n = 0$ to $N_{guesses}$ do
    //initiate the phase $\phi(\omega)$ randomly;
    Sample $\phi(\omega)$ uniformly from $[-\pi, \pi]$
    //transform to real space using inverse Fourier transform
    $g = F^{-1}(p \cdot \mathrm{e}^{i\phi})$
    for $m = 1$ to $N_{inner}$ do
        //appply fourier domain constraints;
        $g_2 = F^{-1}((\alpha p + (1 - \alpha) \mid \hat{g} \mid) \cdot \exp^{i \cdot \phi(\hat{g})})$
        //apply space domain constraints
        $R(x) = 2g_2(x) - g(x)$
        $\beta = \beta_0 + (1 - \beta_0)(1 - \exp(-m/7)^3)$
        $\Omega = \{x : R(x) < 0\} \cup \{x : x \notin [0, s] \times [0, s]\}$
        $g(x) = \begin{cases} \beta_g(x) + (1 - 2\beta)g_2(x), & \texttt{if}\, x \in \Omega, \\ g_2(x), & \texttt{if}\, x \notin \Omega; \end{cases}$
    end for
    $\Omega = \{x : g_2(x) < 0\} \cup \{x : x \notin [0, s] \times [0, s]\}$
    $k_n(x) = \begin{cases} 0, & \texttt{if}\, x \in \Omega, \\ g_2(x), & \texttt{if}\, x \notin \Omega \end{cases}$
end for
Output: $k_n$
```

However, using the phase retrieval method, it cannot guarantee to obtain the same kernel in each of iterations and cannot decide which one is the best blur kernel, as shown in Fig. 1. Figure 1 shows thirty blur kernels which include the symmetric or failure blur kernels after performing thirty iterations in phase retrieval. Assuming that repeating phase retrieval for $n$ times, it can find $n$ blur kernels. Afterwards, an optimal kernel can be obtained from these $n$ kernels. [5] takes the final kernel to deconvolve the blurry image. In our experiments, we utilize a normalized sparsity measure (NSM) [24] to decide the optimal blur kernel from $n$ blur kernels and to estimate the quality of blur kernel.

### *2.3. Measure of blur kernel quality*

As known above, it can obtain $n$ blur kernels after iterating $n$ times. Each of NSM values for deconvolution by using the corresponding kernel can be calculated. Figure 1 shows an example of blur kernel for thirty iteration times. From Fig. 1, it is obvious that the symmetric relationship exists among blur kernels and the estimated blur kernel for each iteration is also different. Hence, the measure of

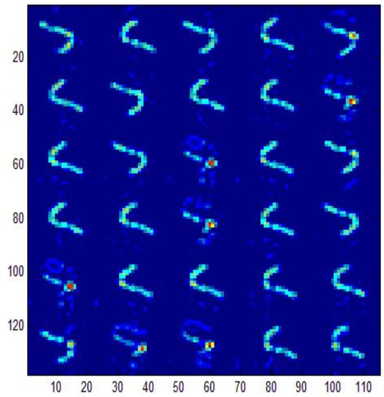
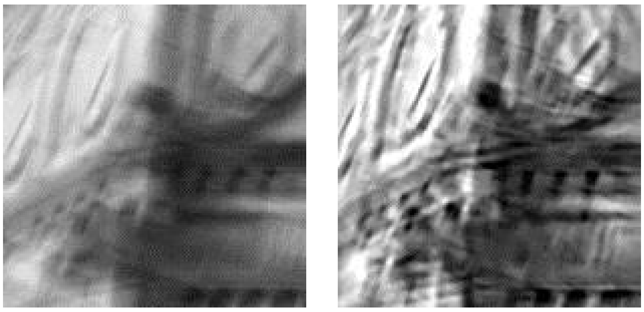


**Figure 1.** Results of iterative phase retrieval for 30 iteration times placed at inverse s-shaped arrangement. Each kernel is of size 21 × 21.



(a) NSM=111.5466 (b) NSM=101.3874

**Figure 2.** NSM value (an example). (a)Blurry image. (b)Reconstructed image.

blur kernel quality will test the symmetry of blur kernel and will make a score. In order to estimate the symmetric characteristic of blur kernel, it has to calculate the NSM score twice. For example, if there are thirty kernels, it will calculate the NSM for sixty times. After computing NSM value, the smaller the NSM score, the better the reconstructed image, as shown in Fig. 2. According to our experimental results, a kernel with the minimum NSM value is a good kernel more confidently.

In our system, we modify the number of computing NSM times and thus speed up an optimal blur kernel acquirement from *n* blur kernels.

### *2.4. Image quality estimation*

Natural image signals are highly structural information, such as pixels exhibiting strong dependencies and containing important information about the structure of the objects in the visual scene. In order to estimate the structural performance of the reconstructed image after deconvolution , we adopt the structural-similarity-based image quality measure (SSIM) [32] instead of the mean squared error (MSE). The SSIM mainly compute the structural similarity between the reference and the distorted signals. However, one usually requires the overall image quality measure, a mean SSIM (MSSIM) derived from SSIM is used to achieve this measure, which can exhibit much better consistency with

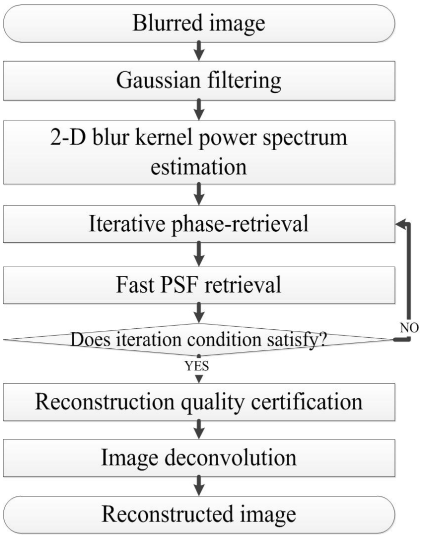

Blurred image
Gaussian filtering
2-D blur kernel power spectrum estimation
Iterative phase-retrieval
Fast PSF retrieval
Does iteration condition satisfy?
NO
YES
Reconstruction quality certification
Image deconvolution
Reconstructed image

**Figure 3.** The diagram of system flowchart.

the qualitative visual appearance. In our experiments, we adopt the MSSIM to estimate the quality of the reconstructed image. The MSSIM index is briefly defined as

$$MSSIM(X,Y) = \frac{1}{M}\sum_{j=1}^{M} SSIM(x_j, y_j), \tag{2.8}$$

where $X$ and $Y$ are the reference and the distorted images, respectively; $x_j$ and $y_j$ are the image contents at the $j$th local window; and $M$ is the number of local windows of the image. The higher the MSSIM, the closer both images.

## 3. Proposed method

In this paper, we propose an image deblurring using fast point spread function (FPSF) method to efficiently and quickly search the optimal blur kernel. Because the method by [5] is time-consuming, in our approach, we will further improve the computational time and speed up decide the optimal blur kernel to deblur the blurred image. Figure 3 illustrates the flowchart of the proposed system. The details of the procedures are described in the following.

### *3.1. Filtering*

In order to improve the blur kernel estimation which is seriously affected by high-frequency components [5], we use Gaussian filter [33] to reduce this influence and improve the blur kernel estimation for a blurred image. Here, we use Gaussian filter with size 5 × 5, $\sigma = 0.6$, and a mean of zero, it is proportional to the size of the neighborhood on which the filter operates. Pixels more distant from the center of the operator have smaller influence. After filtering the blurry image, it can improve the power spectrum of 2D blur kernels, afterwards, the optimal blur kernel can be acquired by our proposed method.

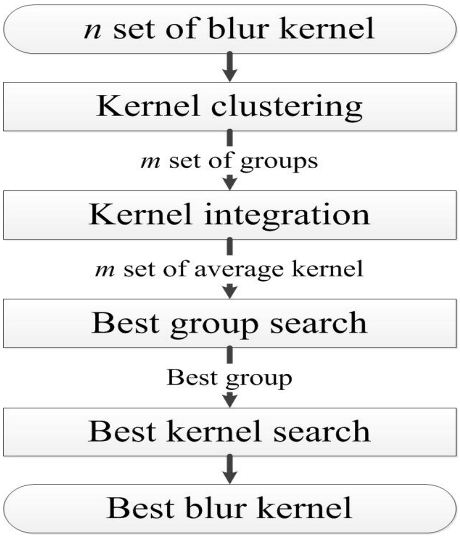


**Figure 4.** The block diagram of the FPSF.

### *3.2. Blur kernel estimation based on FPSF*

After doing blur kernel estimation and iterative phase retrieval described by the previous section, it can obtain the corresponding kernels in each of iterative times in phase retrieval. However, from Figure 1, it is clearly obvious that each of these kernels is different. In other words, it does not guarantee that these kernels can yield a good deblurring result. Based on this reason, we propose a FPSF method to estimate an optimal blur kernel. Using this estimated optimal blur kernel, it can deblur the blurry image by deconvolution. Figure 4 illustrates the flowchart of the FPSF.

This method consists of blur kernel clustering and blur kernel integration. The blur kernel clustering based on MSSIM is to classify these blur kernels obtained by phase retrieval. After performing the clustering, we use the integration to find the optimal kernel from the clusters. For clustering, we use the MSSIM to estimate the similarity of all candidate kernels, kernel which has a high MSSIM will be clustered into the same cluster. The procedures of the clustering method are described as follows.

First of all, assuming that there are $k_n$ blur kernels.

**Step 1)** Select the first kernel in these kernels as an initial base.

**Step 2)** Calculate the MSSIM values for this base and the rest of candidate kernels.

**Step 3)** If MSSIM value is more than or equal to a threshold, then this candidate kernel is classified into the corresponding cluster.

**Step 4)** If MSSIM value is less than a threshold, then a new base is yielded and it will serve as a new cluster.

**Step 5)** Repeat Steps 2 - 4, until all candidate kernels are clustered.

Based on MSSIM characteristic, the higher the MSSIM value, more similar both kernels. According to our experiments, this threshold is set 0.9. An example of kernel clustering algorithm is illustrated as follows. Assuming there are nine blur kernels, they will be clustered.

| Input blur kernel | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $k_1$ | $k_2$ | $k_3$ | $k_4$ | $k_5$ | $k_6$ | $k_7$ | $k_8$ | $k_9$ |

First cycle

| | Candidate kernels | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| base | $k_1$ | $k_2$ | $k_3$ | $k_4$ | $k_5$ | $k_6$ | $k_7$ | $k_8$ | $k_9$ |
| $k_1$ | 1.0 | 0.9 | 0.8 | 0.85 | 0.7 | 0.93 | 0.5 | 0.96 | 0.45 |

| | | | | |
|---|---|---|---|---|
| First cluster ($k_1$) | $k_1$ | $k_2$ | $k_6$ | $k_8$ |

Second cycle

| | Candidate kernels | | | | |
|---|---|---|---|---|---|
| base | $k_3$ | $k_4$ | $k_5$ | $k_7$ | $k_9$ |
| $k_3$ | 1.0 | 0.92 | 0.91 | 0.6 | 0.7 |

| | | | |
|---|---|---|---|
| Second cluster ($k_3$) | $k_3$ | $k_4$ | $k_5$ |

Third cycle

| | Candidate kernels | |
|---|---|---|
| base | $k_7$ | $k_9$ |
| $k_7$ | 1.0 | 0.85 |

| | |
|---|---|
| Third cluster ($k_7$) | $k_7$ |
| Fourth cluster ($k_9$) | $k_9$ |

First, kernel $k_1$ is chosen as a base, next, the MSSIM value is computed. The first cycle can obtain the first cluster called $k_1$ which includes $k_2$, $k_6$, and $k_8$ kernels. Then, kernel $k_3$ is a new cluster. After computing the clustering procedures, all candidate kernels can be classified into the corresponding clusters.

Based on above clustering process, assume there are $n$ blur kernels, they will be classified into $C_m$ kernel clusters where $m \leq n$. As described above, a good kernel can be obtained after computing the NSM for sixty times, but, it is very time-consuming. Therefore, in order to reduce the number of computing NSM times, we propose a blur kernel integration technique to gain this performance. This kernel integration technique consists of mean blur kernel calculation and refining described as follows.

- **Mean blur kernel:** We calculate an average blur kernel corresponding to each of kernel clusters with the relative coordinates. The average blur kernel for each of kernel clusters is defined as

$$K_{m,avg}(x,y) = \frac{1}{g}\sum_{i=1}^{g} C_m(i)(x,y), \tag{3.1}$$

where $g$ denotes the number of kernels corresponding to kernel cluster, $m$ denotes the number of kernel clusters, $C_m(\cdot)(x,y)$ denotes a kernel value corresponding to the $x$ and $y$ coordinates at the $m$th kernel cluster.

- **Refining:** Owing to the difference between kernels in cluster, the average kernel may involve noise (as shown in Figure 5 number 1's result). As described above, the noise will affect the deblurring process, hence, in order to reduce the influence of the noise and keep the important information of the kernel, we further refine the mean blur kernel to gain performance. We make

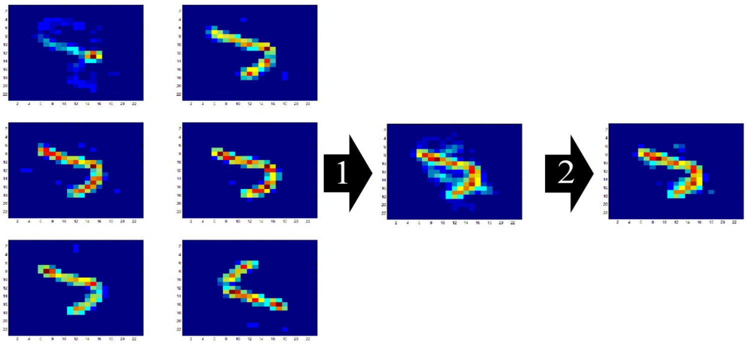


**Figure 5.** Example of kernel integration and its refining in kernel cluster. Number 1: computing an average kernel. Number 2: refining.

a weight matrix to achieve the refining. First, we define a weight matrix of a kernel cluster which represents the number of nonzero kernel values corresponding to the coordinates in the same kernel cluster, the weight matrix is defined as

$$w_m(x,y) = \begin{cases} w_m(x,y)+1, & \texttt{if}\, C_m(i)(x,y) > 0, \\ w_m(x,y), & \texttt{otherwise}, \end{cases} \tag{3.2}$$

where the initial $w_m(x,y)$ sets to zero for $m$ clusters. An average weight value is computed by

$$aveW_m = \frac{\sum_x \sum_y w_m(x,y)}{S_m}, \tag{3.3}$$

where $S_m$ is the number of nonzero kernel values corresponding to the relative cluster. The refining mean blur kernel is expressed as

$$RK_{m,avg}(x,y) = \begin{cases} K_{m,avg}(x,y), \\ \texttt{if}\, w_m(x,y) \geq aveW_m, \\ K_{m,avg}(x,y) = 0, \\ \texttt{otherwise}. \end{cases} \tag{3.4}$$

Figure 5 illustrates an example of kernel integration processing in a kernel cluster.

Based on the blur kernel integration, we can obtain a refining mean blur kernel corresponding to each of kernel clusters. Afterwards, using the measure of blur kernel quality again described by subsection 2.3 for $m$ refining mean blur kernels, we obtain the best mean blur kernel and its corresponding cluster. Because we have recorded the symmetry of the cluster in searching cluster process, thus, we can greatly reduce the computational time and quickly find an optimal blur kernel. The procedures of searching an optimal blur kernel (OBK) are described as follows.

**Step 1)** Give a set of blur kernels which belong to the best kernel cluster, and the corresponding symmetric property of the phase.

**Step 2)** Deconvolve all the blurred images using these kernels.

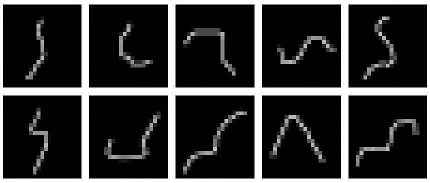


**Figure 6.** Ten blur kernels used to simulate motion-blurred image. The inverse $S$-order denotes these kernels from $k_1$ to $k_{10}$.

**Step 3)** Compute the relative NSM values for all deconvolution results.

**Step 4)** Find the minimum NSM value from these NSM values and its corresponding kernel, this kernel serves as an optimal kernel.

## 4. Experimental results

To verify the performance of our proposed method, the experimental results are compared with Goldstein and Fattal [5] and Krishnan *et al.* [24] to perform the visual quality of the reconstruction image and computational time. Based on the above procedures, we can obtain the optimal kernel for the corresponding blurry image, then each of three components (red, green, blue) for the color image is individually used this optimal kernel to work the non-blind deconvolution method based on [34]; finally, all blurry color images can be reconstructed.

### *4.1. Experimental setup*

All experiments are implemented in Microsoft Visual Studio 2010 C♯, an Intel © i5 dual Core 3.2 GHz, and 4 GB RAM computer with window 7 64 bits platform.

First, we need to make the experimental data by using predefined blur kernels. How to decide the number of predefined blur kernels is a trade-off between objectivity and variety of the experimental data. In order to perform these attributes, in our experiments, we make ten blur kernels with size $21 \times 21$ shown in Figure 6 to simulate ten kinds of blurred images. A database including ten images with size $768 \times 1024$, $682 \times 1024$, $1024 \times 682$, and $720 \times 960$ pixels separated 4, 3, 2, and 1 is given. Then we blurred each of these images using ten blur kernels, it can obtain 100 motion-blurred images for test. In order to avoid noise disturbance to affect kernel estimation, all of experimental data are firstly filtered by Gaussian filter. After filtering, we can obtain more convergent blur kernel by means of our proposed method. Here, we use Gaussian filter with the standard deviation $\sigma = 0.6$ and run thirty iterations for phase retrieval to demonstrate our experiments.

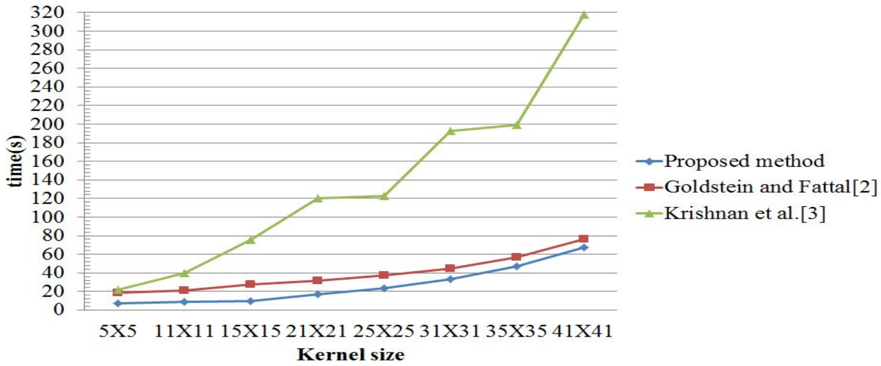


**Figure 7.** The size of blur kernel and its computational time in seconds compared with the proposed method, Goldstein and Fattal's method, and Krishnan *et al.*'s method.

### 4.2. Performance evaluation

For performance measure, we use the peak signal-to-noise ratio (PSNR) and MSSIM to estimate the performance of the system. The PSNR and mean square error (MSE) are defined as

$$\begin{aligned} PSNR &= 10\log\frac{S_{Max}^2}{MSE}, \\ MSE &= \frac{1}{h\times w}\sum_i^h\sum_j^w |I_1(i,j)-I_2(i,j)|^2, \end{aligned} \tag{4.1}$$

where $I_1(i, j)$ and $I_2(i, j)$ denote the reconstructed image and the original one corresponding to the coordinates $(i, j)$. $h$ and $w$ denote the height and width of the image, respectively. For a gray-level image, $S_{Max}$ is 255 gray value.

### 4.3. Experimental results

Because our proposed method improved Goldstein and Fattal method, the power spectrum of blur kernel for each iteration in phase retrieval may be not equal. Hence, in order to obtain the stable performance in our experiments, each blurry image is doing blur kernel estimation and its convolution for ten times. In addition, the size of blur kernel will seriously affect the execution time and reconstructed image quality, therefore, we test the different kernel sizes to demonstrate the execution speed. First of all, we take an average execution time in which the blurry image is doing blur kernel estimation with the different kernel sizes for ten iterations. Figure 7 shows the average execution time for the different blur kernel sizes in our experimental data. From Figure 7, it is clear that the execution time increases with increasing kernel size, and our proposed method is more upgraded than those of methods. Hence, considering the computational time and the visual quality of the reconstructed images, we adopt the blur kernel of size 21 × 21.

Table 1 illustrates the PSNR value and MSSIM value. Image deblurring results of a part of tests are shown in Figures 8–9. Besides man-made test data, we also use the real motion-blurred images to demonstrate our proposed method, as shown in Figures 10–11.

The experimental results have been presented above. For computational time, our proposed method is superior to Goldstein and Fattal's method and Krishnan *et al.*'s method, as shown in Figure 7. For

**Table 1.** The PSNR(dB)/MSSIM values with the different deblur kernels compared with the proposed method, Goldstein and Fattal's [5] method, and Krishnan *et al.*'s [24] method.

| Kernel | Method | | |
|---|---|---|---|
| case | Proposed | [5] | [24] |
| $k_1$ | 22.13/0.94 | 22.15/0.81 | 20.47/0.70 |
| $k_2$ | 23/0.96 | 23.24/0.85 | 20.36/0.73 |
| $k_3$ | 23.95/0.96 | 24.36/0.53 | 25.94/0.85 |
| $k_4$ | 22.6/0.95 | 22.47/0.83 | 22.04/0.79 |
| $k_5$ | 21.69/0.93 | 21.48/0.79 | 20.16/0.65 |
| $k_6$ | 22.42/0.94 | 22.49/0.82 | 22.8/0.83 |
| $k_7$ | 19.43/0.88 | 19.33/0.68 | 18.77/0.57 |
| $k_8$ | 24.27/0.95 | 24.28/0.95 | 21.16/0.88 |
| $k_9$ | 22.13/0.94 | 22.14/0.80 | 21.63/0.78 |
| $k_{10}$ | 23.4/0.95 | 23.54/0.82 | 20.23/0.66 |
| Ave. | 22.5/0.94 | 22.55/0.82 | 21.36/0.75 |

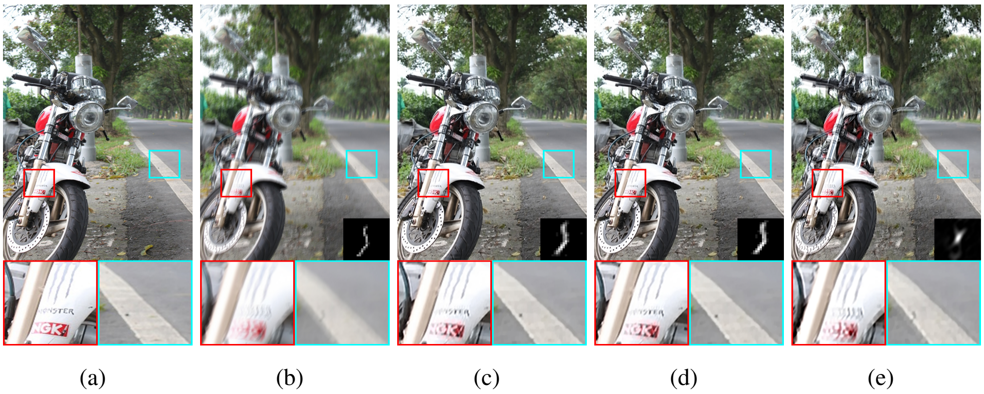
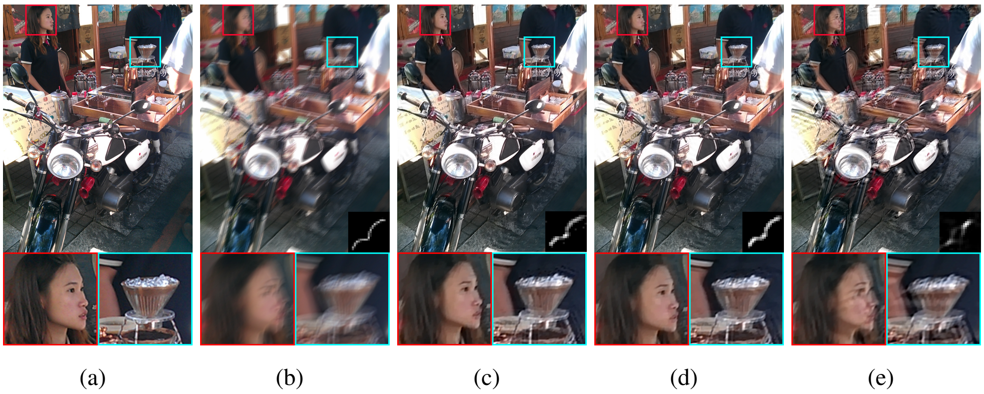
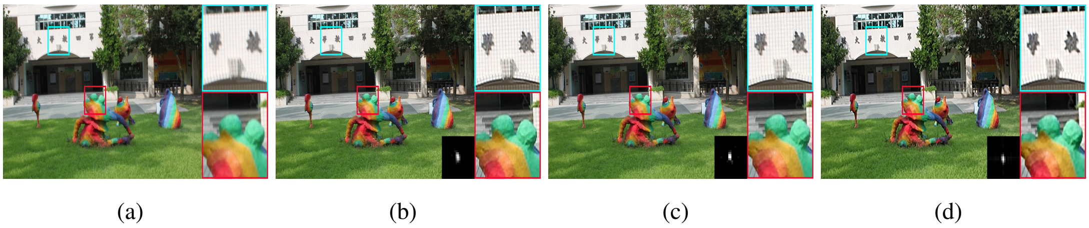


**Figure 8.** Deblurring results. (a)Original image, (b)blurred image generated by using $k_1$ kernel located at right-bottom, (c)our proposed method, PSNR=21.87dB, MSSIM=0.94, (d)Goldstein and Fattal's method, PSNR=21.50dB, MSSIM=0.81, and (e)Krishnan *et al.*'s method, PSNR=19.88dB, MSSIM=0.70.



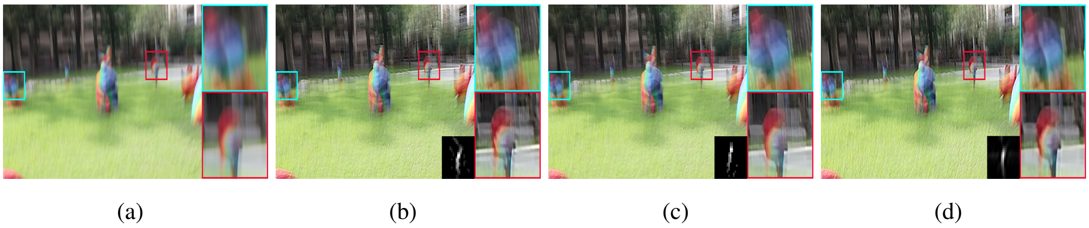
(a) (b) (c) (d) (e)

**Figure 9.** Deblurring results. (a)Original image, (b)blurred image generated by using $k_8$ kernel located at right-bottom, (c)our proposed method, PSNR=23.65dB, MSSIM=0.95, (d)Goldstein and Fattal's method, PSNR=23.39dB, MSSIM=0.95, and (e)Krishnan *et al.*'s method, PSNR=20.07dB, MSSIM=0.89.



(a) (b) (c) (d)

**Figure 10.** Deblurring results for real motion-blurred image (case 1). (a)Original image, (b)our proposed method, (c)Goldstein and Fattal's method, and (d)Krishnan *et al.*'s method.



(a) (b) (c) (d)

**Figure 11.** Deblurring results for real motion-blurred image (case 2). (a)Original image, (b)our proposed method, (c)Goldstein and Fattal's method, and (d)Krishnan *et al.*'s method.

MSSIM and PSNR values, our proposed method is superior to Krishnan *et al.*'s method and is close to Goldstein and Fattal's method, illustrated in Table 1. For real blurry images, because many factors caused the motion-blurred images are unknown beforehand, the recovered image still existed the failure, as shown in Figure 11. In summary, the global performance for MSSIM and computational time is superior to that for the methods of Goldstein and Fattal and Krishnan *et al.*

## 5. Conclusion

In this paper, we have proposed an image deblurring based on FPSF and clustering to recover the sharp image. Based on FPSF method applied the intelligence computing on the estimated image, the advantage of the computing strategy could reduce much more the computational complexity, at the same time, the optimal blur kernel could be estimated efficiently. As the experimental results, our proposed algorithm could efficiently reduce computational time. Also, for the blurry images, the proposed algorithm could restore the images with a good visual quality. It could enhance the image quality shown in many kinds of the view devices.

## Conflict of interest

There is no conflict of interest in this paper.

## References

1. M. Egmont-Petersen, D. De Ridder and H. Handels, Image processing with neural networks a review, *Pattern Recogn.*, **35** (2002), 2279–2301.
2. D. De Didder, R. P. Duin, M. Egmont-Petersen, et al., Nonlinear image processing using artificial neural networks, *Adv. Imag. Elect. Phys.*, **126** (2003), 352–450.
3. C. J. Schuler, M. Hirsch, S. Harmeling, et al., *IEEE T. Pattern Anal.*, **38** (2016), 1439–1451.
4. A. Levin, Blind motion deblurring using image statistics, *Proc. Adv. Neural Infor. Process. Sys.*, (2006), 841–848.
5. A. Goldstein and R. Fattal, Blur-kernel estimation from spectral irregularities, European Conference on Computer Vision, (2012), 622–635.
6. W. Hu, J. Xue and N. Zheng, PSF estimation via gradient domain correlation, *IEEE T. Image Process.*, **21** (2012), 386–392.
7. H. Y. Huang and W. C. Tsai, Blurred image restoration using fast blur-kernel estimation, 10th International Conference on Intelligent Information Hiding and Multimedia Signal Processing, (2014), 435–438.
8. Y. Liao, W. Li, J. Cui, et al., Blur kernel estimation model with combined constraints for blind image deblurring, Digital Image Computing: Techniques and Applications, (2018), 1–8.
9. W. H. Richardson, Bayesian-based iterative method of image restoration, *J. Opt. Soc. Am.*, **62** (1972), 55–59.
10. Y. W. Tai, P. Tan and M. S. Brown, Richardson-Lucy deblurring for scenes under a projective motion path, *IEEE T. Pattern Anal.*, **33** (2011), 1603–1618.

11. H. L. Yang, P. H. Huang and S. H. Lai, A novel gradient attenuation Richardson-lucy algorithm for image motion deblurring, *Signal Process.*, **103** (2014), 399–414.

12. M. Welk, P. Raudaschl, T. Schwarzbauer, et al., Fast and robust linear motion deblurring, *Signal Image Video P.*, **9** (2015), 1221–1234.

13. M. Dobeš, L. Machala and T. Fürst, Blurred image restoration:A fast method of finding the motion length and angle, *Digit. Signal Process.*, **20** (2010), 1677–1686.

14. G. Liu, S. Chang and Y. Ma, Blind image deblurring using spectralproperties of convolution operators, *IEEE T. Image Process.*, **23** (2014), 5047–5056.

15. T. Yue, S. Cho, J. Wang, et al., Hybrid image deblurring by fusing edge and power spectrum information, Process of European Conference on Computer Vision, (2014).

16. A. M. Deshpande and S. Patnaik, Single image motion deblurring: An accurate PSF estimation and ringing reduction, *Optik*, **125** (2014), 3612–3618.

17. W. A. Shao, Q. Ge, H. S. Deng, et al., Motion deblurring using non-stationary image modeling, *J. Math. Imaging Vis.*, **52** (2015), 234–248.

18. N. He, K. Lu, B. K. Bao, et al., Single-image motion deblurring using an adaptive image prior, *Inform. Science*, **281** (2015), 736–749.

19. J. Jia, Single image motion deblurring using transparency, Process of IEEE Conference on Computer Vision and Pattern Recognition, (2007), 1–7.

20. A. Levin, Y. Weiss, F. Durand, et al., Understanding and evaluating blind deconvolution algorithms, Process of IEEE Conference on Computer Vision and Pattern Recognition, (2009), 1964–1971.

21. A. Levin, Y. Weiss, F. Durand, et al., Understanding blind deconvolution algorithms, *IEEE T. Pattern Anal.*, **33** (2011), 2354–2367.

22. L. Xu, S. Zheng and J. Jia, Unnatural $L_0$ sparse representation for natural image deblurring, Process of IEEE Conference on Computer Vision and Pattern Recognition, (2013).

23. T. S. Cho, C. Paris, B. K. P. Horn, et al., Blur kernel estimation using the Radon transform, Process of IEEE Conference on Computer Vision and Pattern Recognition, (2011), 241–248.

24. D. Krishnan, T. Tay and R. Fergus, Blind deconvolution using a normalized sparsity measure, Process of IEEE Conference on Computer Vision and Pattern Recognition, (2011), 223–240.

25. M. Ljubenović and M. A. T. Figueiredo, Blind image deblurring using class-adapted image priors, Process of IEEE Conference on Image Processing, (2017), 490–494.

26. J. Pan, D. Sun, H. Pfister, et al., Deblurring images via dark channel prior, *IEEE T. Pattern Anal.*, **10** (2018), 2315–2328.

27. D. J. Field, Relations between the statistics of natural images and the response properties of cortical cells, *J. Opt. Soc. Am. A*, **4** (1987), 2379–2394.

28. G. J. Burton and I. R. Moorhead, Color and spatial structure in natural scenes, *Appl. Optics*, **26** (1987), 157–170.

29. L. Cohen, The generalizaton of the Wiener-Khinchin theorem, *IEEE Signal Proc. Let.*, **5** (1998), 292–294.

30. J. R. Fienup, Phase retreival algorithms: A comparsion, *Appl. Opt.*, **21** (1982), 2758–2769.

31. S. Watanabe, H. Shioya and K. Gohara, Phase retrieval based on an evolution multicriterion optimisation method, Process of IEEE Conference on Evolutionary Computation, (2012), 1–8.

32. Z. Wang, A. C. Bovik, H. R. Sheikh, et al., Image quality assessment: from error visibility to structural similarity, Process of IEEE Conference on Image Processing, **13** (2004), 600–612.

33. M. Sonka, V. Hlavac and Roger Boyle, Image processing, Analysis, and Machine Vision, 3rd Ed., Thomson Learning Pub., 2008.

34. D. Krishnan and R. Fergus, Fast image deconvolution using hyper-Laplacian priors, *Proc. Conf. NIPS*, (2009), 1033–1041.

AIMS Press