



Joint blur kernel estimation and CNN for blind image restoration

Liqing Huang, Youshen Xia*

College of Mathematics and Computer Science, Fuzhou University, China

ARTICLE INFO

Article history:

Received 30 March 2018

Revised 22 November 2018

Accepted 27 December 2018

Available online xxx

Keywords:

Blind image restoration

Blur kernel

Blur support parameter estimation

Blur type identification

CNN

ABSTRACT

Convolutional neural networks (CNN) have shown its excellent performance in computer vision fields. Recently, they are successfully applied to image restoration. This paper proposes a joint blur kernel estimation and CNN method for blind image restoration. The blur kernel estimation is based on both blur support parameter estimation and blur type identification. An automatic feature line detection algorithm is presented for blur support parameter estimation and a dictionary learning algorithm is presented for the blur type identification. Once the blur kernel estimate is obtained, we use an effective CNN for iterative non-blind deconvolution, which is able to automatically learn image priors. Compared with current blind image restoration methods, the proposed joint method can obtain restored images under three types of unknown blur kernels. The experimental result shows that the proposed blur kernel estimation algorithm can provide high accuracy results. Furthermore, the proposed joint blur kernel estimation and CNN algorithm is superior to conventional blind image restoration algorithms in terms of restoration quality and computation time.

© 2019 Elsevier B.V. All rights reserved.

1. Introduction

Image restoration has been widely applied for medical imaging, astronomical imaging, remote sensing, microscopy imaging, photography deblurring, and among others [1,2]. Conventional image restoration techniques include the Wiener and Kalman Filtering technique, the regularization technique, the variational regularization technique, joint statistical regularization technique, and the fusion regularization technique under the assumption that the blur kernel is known [3–6]. Recently, it was reported that convolutional neural networks can be well applied to non-blind image restoration and super-resolution [7–16]. For example, Schuler et al. presented a two-step procedure based on a neural network [7]. Xu et al. [8] proposed outlier-rejection deconvolution CNN. Zhang et al. [9] proposed a CNN denoisers-based method for image denoising. Zhang et al. [16] proposed a fully convolutional network for iterative non-blind deconvolution in a two-stage framework.

Because the blur kernel is often unknown in practice, blind image deconvolution is required. Various blind image deconvolution techniques have been presented in recent decades. They can be mainly divided into three class methods. The first is simultaneously to estimate the blur kernel and to restore the image [17–23,30,31], mainly including the iterative blind deconvolution (IBD) algorithm [17], the nonnegative and support constraints recursive inverse

Filtering (NAS-RIF) algorithm [20], the double regularization approach [23] and the total variation approach [18,22], and the soft double regularization approach [31]. Recently, Pan et al. [24] present a blind image deblurring method based on the dark channel prior. This work is inspired by the interesting observation that the dark channel of blurred images is less sparse. Yan et al. [25] proposed a blind image deblurring method based on both bright channel prior and dark channel prior. For this type of method, the alternating minimization iteration technique is required [30].

The second is the parametric method that first estimates the blur kernel from the received image and then performs a non-blind image restoration algorithm to obtain the restored image [26,27]. The classic ARMA parameter estimation method has a local minimum problem since its cost function is nonconvex [28]. An effective ARMA parameter estimation method was developed by overcoming the nonconvex problem [29]. The ARMA parameter estimation method is suitable for general blur kernels but the blur support parameter is required in advance. In recent decade, several effective estimation methods for blur support parameters were proposed [26,30–36]. Among them, the discrete spatial estimation-based method [31,32] was presented for estimating approximately Gaussian and uniform blur kernels. Xue and Blu proposed the blur-SURE estimation-based method for Gaussian blur kernel [26]. Oliveira et al. proposed the power-spectrum estimation method for linear motion and Out-of-Focus blur kernels [60]. Liu et al. proposed a learning-based method for estimating the motion blur kernel size [33]. Li et al. proposed a deep

* Corresponding author.

E-mail address: ysxia@fzu.edu.cn (Y. Xia).

learning-based method [34] and Xu et al. proposed motion blur kernel estimation method based on deep learning [35]. These learning-based methods were reported to be very effective on estimating camera-shake motion blur. On the other hand, accurately identifying blur type is also of significant topic in blind image deconvolution since the type of blur is usually unknown [41]. Several blur type identification methods were presented [37–42]. For example, Liu et al. [37] introduced a Bayesian classifier by using local autocorrelation congruency. Su et al. [38] presented an alpha channel feature-based classification method. Palm et al. [40] introduced a convolutional neural network for blur classification. Rugna and Konik proposed [43] a automatic segmental technique to discriminate blurry regions. Recently, Yan and Shao proposed [41] a deep learning method for blur kernel identification. Wang et al. proposed an motion blur parameter estimation method by using frequency spectrum image information [75].

The third is the learning-based method [41,46–52]. Schuler et al. [46] presented a deep layered architecture that is trained end-to-end on a set of artificially generated training examples. Wieschollek et al. [47] presented a neural network approach for multi-frame blind deconvolution. The discriminative approach combines a deconvolution filter and non-trivial averaging of Fourier coefficients. Nah et al. [48] proposed a multi-scale convolutional neural network without assuming any restricted blur kernel model. Svoboda et al. [49] used a convolutional neural networks for motion deblurring and Hradi et al. [50] used a convolutional neural network for direct text deblurring. Yan et al. [41] introduced deep learning method for blur kernel estimation. Chakrabart [51] presented a new learning method for blind motion deblurring, where a neural network is trained to compute estimates of sharp image patches from observations that are blurred by an unknown motion kernel. By using RBF neural network, Dash and Majhi presented an motion blur parameter estimation method for image restoration [74]. Lai et al. [52] gave a comprehensive study for single image blind deblurring.

This paper proposes a blind image restoration method based on joint blur kernel estimation and CNN. The main works of our paper are two fold. We present a blur support parameter estimation method and a blur type identification method for three types of blur kernels. An automatic feature line detection algorithm is presented for the blur support parameters and a dictionary learning algorithm is presented for the blur type identification. Once the blur kernel is estimated, we use an effective CNN for iterative non-blind deconvolution. Compared with current blur kernel estimation method, the proposed blur kernel estimation method can estimate fast the blur support parameter and accurately identify the blur type. As a result, the proposed joint blur kernel estimation and CNN method can enhance the quality of restored images under three types of unknown blur kernels. The experimental result shows the effectiveness of the proposed blur parameter estimation algorithm. Furthermore, the proposed joint algorithm is superior to conventional blind image restoration algorithms in terms of both restoration quality and computation time.

The rest of this paper is organized as follows. In Section 2, we briefly describe image degradation model, three types of blur kernel models, and CNN. In Section 3, a joint blur kernel estimation and CNN method for blind image restoration is introduced. In Section 4, experimental results confirm the effectiveness of the proposed joint algorithm. Section 5 concludes the paper.

2. Image degradation model and CNN

2.1. Convolutional degradation model

We are concerned with the following linear degraded image model:

$$y(i, j) = x(i, j) * h(i, j) + n(i, j) \quad (i, j) \in \Omega \quad (1)$$

where (i, j) is the discrete pixel coordinate of the image, $x(i, j)$ is the original image, $\Omega \in R^2$ is the support of the image, $y(i, j)$ is the degraded image, $h(i, j)$ is the blur kernel or the point spreading function (PSF), $*$ is the discrete 2-D linear convolution operator, and $n(i, j)$ is the additive white Gaussian noise.

2.2. Three types of blur kernel models

Typical examples of the parametric prior blur model can be found in the applications of linear motion blur [34,35], out-of-focus blur [26,41], and atmospheric turbulence blur [27,53]. We thus study the following three types of blur kernel models.

For long-term exposure through atmosphere, Gaussian blur model is used:

$$h(i, j) = Q * e^{-\frac{(i^2+j^2)}{2\sigma^2}} \quad (i, j) \in S_h \quad (2)$$

where (i, j) denotes the location of pixel, Q is a normalization constant such that $\sum_{i,j} h(i, j) = 1$, σ^2 is the Gaussian variance, and S_h is the support parameter of the blur. This type of blur is common in remote sensing and satellite imaging.

The out-of-focus blur is usually used as the model for photographic defocusing [54], defined as:

$$h(i, j) = \begin{cases} \frac{1}{\pi r^2} \sqrt{i^2 + j^2} \leq r \\ 0, \quad \text{otherwise} \end{cases} \quad (3)$$

where r denotes the parameter of the out-of-focus blur and $2r + 1$ is called the blur support parameter.

If the camera movement or object motion is fast relative to the exposure period, we may approximate it as a linear motion blur [34]:

$$h(i, j) = \begin{cases} \frac{1}{L} & (i, j) \left(\frac{\sin(\omega)}{\cos(\omega)} \right) = 0, i^2 + j^2 \leq L^2/4 \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

where ω is the motion direction with its angle to x axis, and L is the length of motion in pixels. An example of the linear motion blur is a horizontal motion blur given by:

$$h(i, j) = \begin{cases} \frac{1}{L} & -L/2 \leq i \leq L/2, \quad j = 0 \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

A neural extension of the linear motion blur is the uniform 2-D blur given by:

$$h(i, j) = \begin{cases} \frac{1}{LM} & -L/2 \leq i \leq L/2, \quad -M/2 \leq j \leq M/2 \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

where L and M are called the blur support parameters. The uniform 2-D blur is sometimes used as a cruder approximation to the out-of-focus blur, and used in super-resolution restoration.

The goal of this paper is to propose a blind image restoration method under the three types of unknown blur kernels mentioned above.

2.3. CNN model

Convolutional neural network (CNN) is a kind of machine learning method with deep supervised learning. Recently, CNN has shown excellent performance in various computer vision tasks, such as image classification, image restoration, and image super-resolution. For non-blind image restoration, the CNN is used for

solving the image deblurring problem by training a network architecture that minimizes:

$$\frac{1}{2|N|} \sum_{i \in N} \|f(y_i) - x_i\|^2 \quad (7)$$

where $|N|$ is the number of training degraded-clear image pairs (y_i, x_i) and $f(\cdot)$ is the mapping learned by CNN. Recently, Zhang et al. [9] proposed using a seven-layer CNN for image restoration:

$$L_k = \sigma(W_k * L_{k-1} + b_{k-1}) \quad (k = 1, \dots, 7); \quad L_0 = y \quad (8)$$

where $\sigma(\cdot)$ is the nonlinear active function (sigmoid or hyperbolic tangent), W_k is the weight mapping the $(k-1)$ th layer to the k th one and b_{k-1} is the vector value bias.

3. Proposed method for blind image restoration

Most of conventional blind image restoration methods all assume that the blur support parameter or blur type is known in advance. Various blur kernel estimation-based methods and the learning-based methods for blind image restoration were presented. Although it is reported that these methods are effective, fast and accurate blur kernel estimation for blind image restoration remains a challenging issue.

This paper proposes a fast and effective blind image restoration method based on joint blur kernel estimation and CNN. The blur kernel estimation includes both blur support parameter estimation and blur type identification.

3.1. Support parameter estimation

Since linear motion blur may be viewed as a special case of uniform blur, we focus on estimating the support parameter estimation of both Gaussian blur and uniform blur kernels.

Recently it was reported in [41] that the logarithmic power spectrum can describe clearer edge information than the edge detector technique [55]. Inspired by this fact, we propose an automatic feature line estimation-based method for fast and accurately estimating the support parameter estimation of both Gaussian blur and uniform blur kernels. Consider the degraded image $g \in R^{m \times n}$. According to the logarithmic spectrum, we compute the normalized logarithm of Fourier transform of g , and then compute the binary transform matrix produced by using *edge* function. Our feature line estimation-based method is to calculate the number of the feature line in the binary feature matrix, where each feature line has the pixel value being 1. If the pixel values of each row in binary feature matrix are all 1, the feature line is easily recognized. However, the pixel values of each row are not so under strong noise environments. To overcome this difficulty, we recognize the feature line at each row by using a positive integer number such as $n/2$ where the number of pixel values being 1 is greater than it. Once recognizing some row as a feature line, we store the current row number into row set ind_1 . Similarly, we recognize the feature line at each column by using a positive integer number, such as $m/2$ where the number of pixel values being 1 is greater than it. Once recognizing some row as a feature line, we store the current column number into column set ind_2 .

Algorithm 1 summarizes the proposed feature line estimation-based method for support parameter estimation. Fig. 1 displays the block diagram of the proposed **Algorithm 1**, where sampling interpolation is used to enhance feature line detection for improving the estimation accuracy, *edge* function is employed in MATLAB, and the normalized logarithm is given in [41].

It is seen that major computation of **Algorithm 1** lies in step 1 and step 3 for computing binary feature matrix. Having low computational complexity and no network parameter learning, the proposed **Algorithm 1** has a fast computational speed. Moreover,

Algorithm 1: for blur support parameter.

- Input :** Given $m \times n$ degraded image g and design parameters $\tau > 0$ and $\varepsilon > 0$. Let G denote Fourier transform of g , and let initial support parameter be $hsize_1 \times hsize_2 = 1 \times 1$;
- Output:** The blur support of degraded image g :
- 1 Compute the normalized logarithm of G :
 - ($\log(|G|)_{norm} = \frac{\log(|G|) - \log(|G_{min}|)}{\log(|G_{max}|) - \log(|G_{min}|)}$)
 - where $G_{min} = \min(G(:))$; $G_{max} = \max(G(:))$;
 - 2 Up-sampling the normalized logarithm image $\log(|G|)_{norm}$ using *imresample* function;
 - 3 Compute binary feature matrix $J = \text{edge}(\log(|G|)_{norm})$ using *edge* function;
 - 4 Calculate the number of pixel value being 1 at each row of binary feature matrix J . If the number is large than $n/2$ at some row, then the row is recognized as a feature line and save the row number into the row set ind_1 ;
 - 5 Calculate the number of pixel value being 1 at each column of binary feature matrix J . If the number is large than $m/2$ at some column, then the column is recognized as a feature line and save the column number into the column set ind_2 ;
 - 6 Pre-processing boundary feature lines:
 - 7 if $ind_1(j) < \tau$ and $ind_1(j) < \tau$, then $ind_1 = j$.
 - 8 if $ind_2(j) < \tau$ and $ind_2(j) < \tau$, then $ind_2 = j$.
 - 9 Output support parameter $hsize_1 \times hsize_2$. If $hsize_1 \neq hsize_2$, then uniform motion blur is identified. Otherwise, Gaussian blur is identified;
-

because the logarithmic power spectrum can provide clearer edge information, the proposed **Algorithm 1** can obtain a high accurate support parameter estimate.

3.2. Blur type identification

In Fig. 2(a) and (e) are original images and Fig. 2 (b) and (f) are two degraded images. Fig. 2(c) and (g) are two restored images under Gaussian blur type and Fig. 2(d) and (h) are two restored images under motion blur type. It is seen that accurate blur type identification is of important significance for blind image restoration.

By combining the logarithmic power spectrum [61], the log-Gabor filter [45], and the feature similarity index [44] together, we propose a dictionary learning-based method for blur type identification. Instead of deep neural network learning, our dictionary learning algorithm uses the structural similarity index so as to identify the best structural similarity between observed images and dictionary images. Fig. 3 displays the block diagram of the dictionary learning algorithm, which mainly includes computing the normalized logarithm of blurred images from both observed image and dictionary images, the structural similarity index, and the best structural similarity.

More exactly, similar to constructing dictionary library [56], there are r_1 original images and r_2 blur kernels with r_3 blur sizes. This dictionary library contains $r_1 \times r_2 \times r_3$ blurred images and their logarithmic normalized images. Then the dictionary library set is given by $S = \{S_k\}_{1}^{r_1 \times r_2 \times r_3}$ where S_k is the logarithmic normalized image. For each S_k , from [43,44] we first compute the gradient magnitude of S_k given by:

$$D(S_k) = \sqrt{(\Delta_x S_k)^2 + (\Delta_y S_k)^2} \quad (9)$$

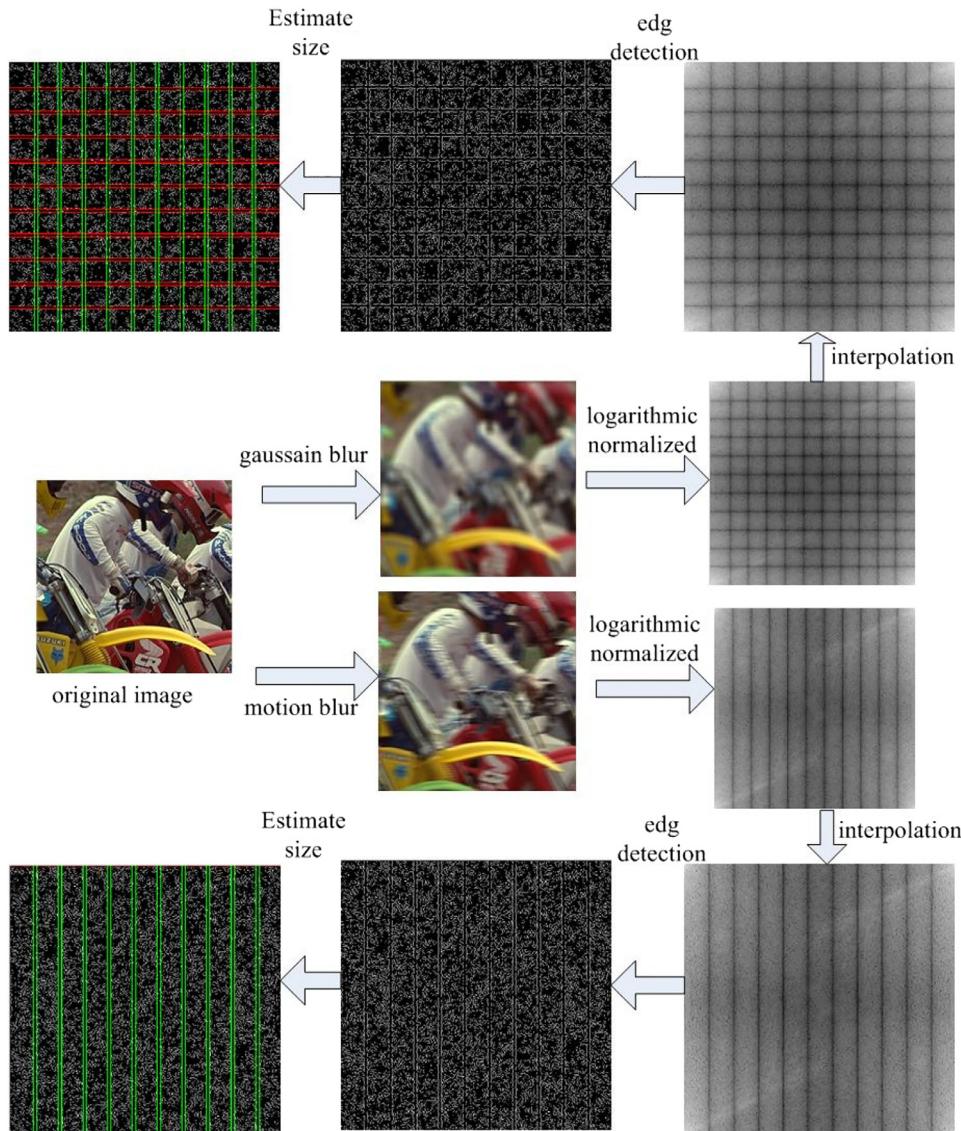


Fig. 1. Block diagram of proposed support parameter estimation.

where the partial derivatives $\Delta_x S_k$ and $\Delta_y S_k$ of the image S_k along horizontal and vertical directions are given by:

$$\Delta_x S_k = \frac{1}{16} \begin{bmatrix} 3 & 0 & -3 \\ 10 & 0 & -10 \\ 3 & 0 & -3 \end{bmatrix} \otimes S_k,$$

$$\Delta_y S_k = \frac{1}{16} \begin{bmatrix} 3 & 10 & 3 \\ 0 & 0 & 0 \\ -3 & -10 & -3 \end{bmatrix} \otimes S_k$$

We then compute the phase congruency matrix of S_k defined as:

$$PC(S_k) = \frac{\sum_j E_{\theta_j}(S_k)}{\varepsilon + \sum_n \sum_j A_{n,\theta_j}(S_k)} \quad (10)$$

where $E_{\theta_j}(S_k) = \sqrt{(\sum_n e_{n,\theta_j}(S_k))^2 + (\sum_n o_{n,\theta_j}(S_k))^2}$, $A_{n,\theta_j}(S_k) = \sqrt{e_{n,\theta_j}(S_k)^2 + o_{n,\theta_j}(S_k)^2}$ and $e_{n,\theta_j}(S_k) = \text{real}(ifft2(SP_k))$, $o_{n,\theta_j}(S_k) = \text{imag}(ifft2(SP_k))$, $P_k = fft2(S_k) \cdot P(\omega, \theta_j)$ and $P(\omega, \theta_j)$ is the 2-D log-Gabor filter defined in [45].

The proposed dictionary learning algorithm is described in **Algorithm 2**. Because we joint three popular techniques: the logarithmic power spectrum, the log-Gabor filter, and the feature

similarity index, **Algorithm 2** can identify the blur type at a high accuracy rate.

3.3. Joint blur kernel estimation and CNN-based algorithm

Recently, it was reported in [8,9,57,58] that the CNN was successfully applied to non-blind image restoration. In this section, for blind image restoration we combine the proposed blur kernel estimation and the CNN designed in [9] together. Image restoration is an inverse problem. Conventional regularization methods can minimize a constrained optimization problem:

$$\hat{x} = \arg \min_x \frac{1}{2} \|Hx - y\|_2^2 + \lambda \Phi(x) \quad (11)$$

where $\Phi(x)$ is the regularization term and $\lambda > 0$ is the regularization parameter. Instead of (11), the CNN is here based on the half-quadratic splitting (HQS) framework. HQS method is to minimize:

$$L_\mu(x, z) = \frac{1}{2} \|Hx - y\|_2^2 + \lambda \Phi(z) + \frac{\mu}{2} \|z - x\|_2^2 \quad (12)$$

where μ is a penalty parameter which varies iteratively in a non-descending order. To minimize (12), the alternate iteration scheme

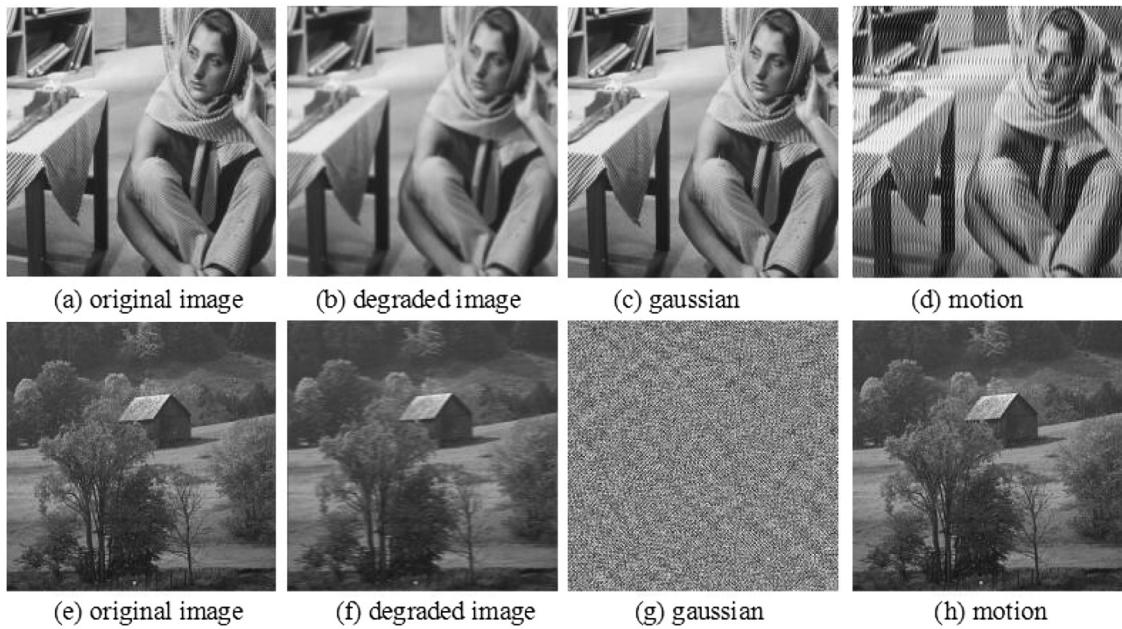


Fig. 2. The influence of blur type identification on blind image restoration.

Training set of blurred image's log normalization

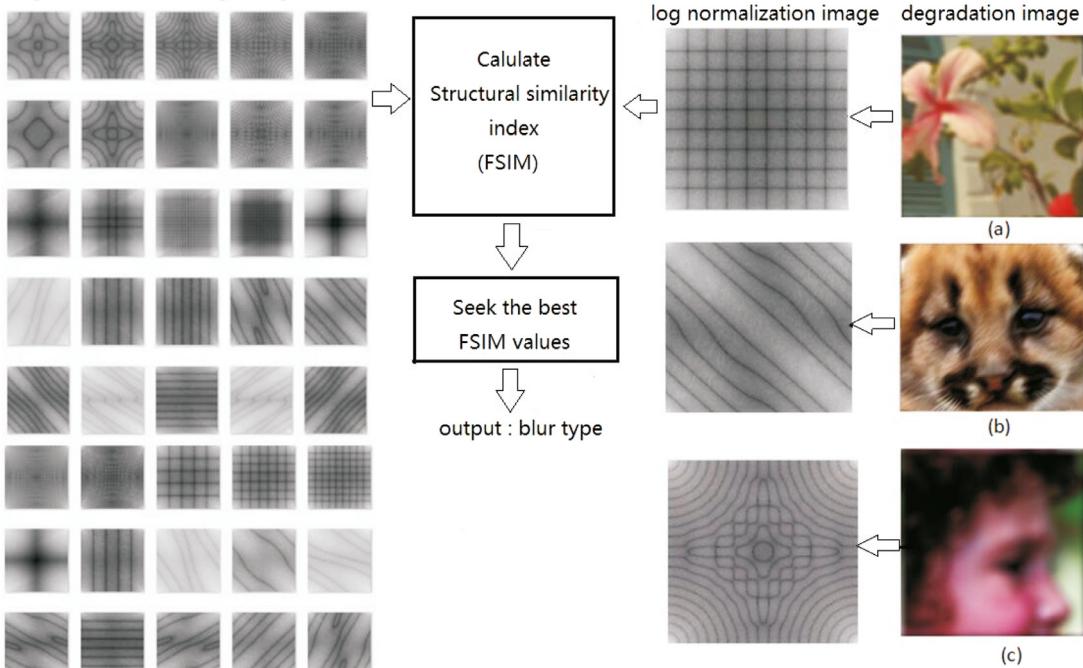


Fig. 3. Block diagram of proposed blur type identification.

is given by:

$$\begin{cases} x^{k+1} = \underset{\tilde{x}}{\operatorname{argmin}} \|H^T x - y\|_2^2 + \mu \|x - z^k\|_2^2 \\ z^{k+1} = \underset{\tilde{z}}{\operatorname{argmin}} \frac{\mu}{2} \|z - x^{k+1}\|_2^2 + \lambda \Phi(z) \end{cases} \quad (13)$$

That is,

$$\begin{cases} x^{k+1} = (H^T H + \mu I)^{-1} (H^T y + \mu z^k) \\ z^{k+1} = \underset{z}{\operatorname{argmin}} \frac{1}{2(\sqrt{\lambda/\mu})^2} \|z - x^{k+1}\|_2^2 + \lambda \Phi(z) \end{cases} \quad (14)$$

By using x^{k+1} as input of the CNN, we minimize the following loss function of residual image:

$$\ell(\Theta) = \frac{1}{2N} \sum_{i=1}^N \|f(y_i; \Theta) - (y_i - x_i)\|_F^2 \quad (15)$$

where $\{y_i, x_i\}$ represents N blur-clean patch pairs. To optimize the network parameters Θ , the Adam solver [59] is adopted. The architecture of the CNN consists of seven layers with three different blocks as shown in [9], where "Dilated Convolution+ReLU" block used in the first layer, "Dilated Convolution+Batch Normalization+ReLU" blocks used in the second to sixth layers, and "Dilated Convolution" block used in the last layer. The dilation factors

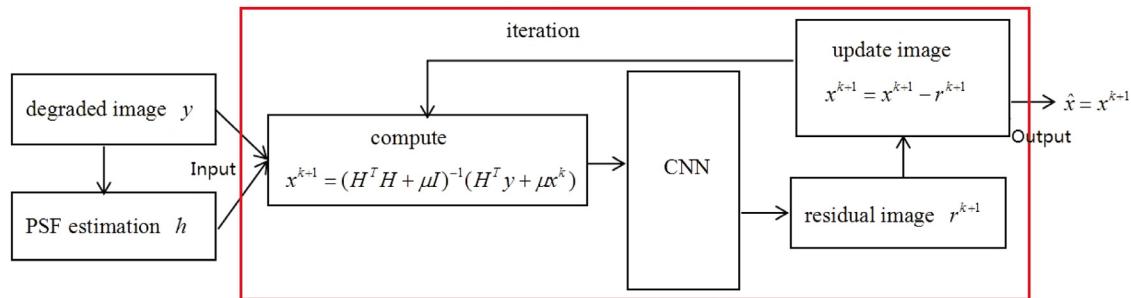


Fig. 4. Block diagram of Algorithm 3.

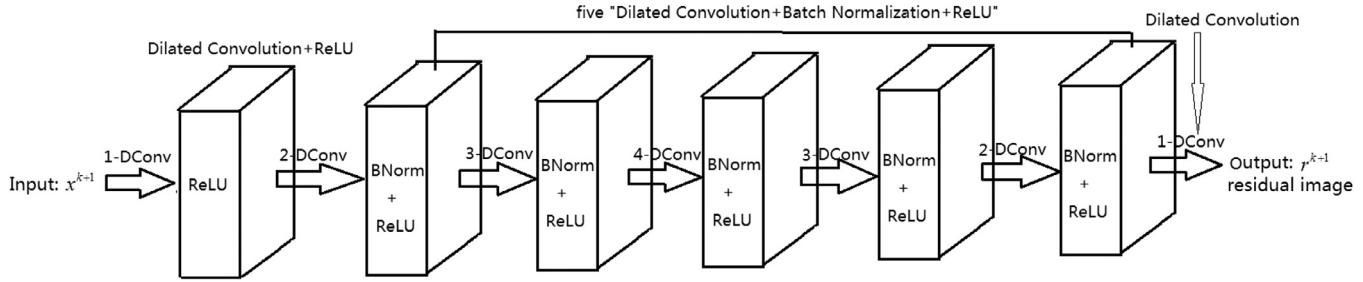


Fig. 5. Architecture of CNN.

Algorithm 2: for blur type identification.

Input : Given $m \times n$ degraded image g and design parameters $T_1 > 0$ and $T_2 > 0$. Let G denote Fourier transform of g ;

Output: The type of degraded image g ;

1 Compute the normalized logarithm of G by:

$$G_{log} = (\log(|G|))_{norm} = \frac{\log(|G|) - \log(|G_{min}|)}{\log(|G_{max}|) - \log(|G_{min}|)}$$

where $G_{min} = \min(G(:))$; $G_{max} = \max(G(:))$.

2 For each image S_k in the dictionary and G_{log} , compute FSIM value:

$$F(G_{log}, S_k) = \frac{\sum_{G_{log} \in \Omega} S_L(G_{log}, S_k) \cdot PC^*(G_{log}, S_k)}{\sum_{G_{log} \in \Omega} PC^*(G_{log}, S_k)}$$

where $S_L(G_{log}, S_k) = S_{PC}(G_{log}, S_k) \cdot S_G(G_{log}, S_k)$, $PC^* = \max(PC(G_{log}), PC(S_k))$, $S_{PC}(G_{log}, S_k) = \frac{2PC(G_{log}) \cdot PC(S_k) + T_1}{PC(G_{log}) \cdot PC(S_k) + T_1}$, $S_G(G_{log}, S_k) = \frac{2D(G_{log}) \cdot D(S_k) + T_2}{D(G_{log}) \cdot D(S_k) + T_2}$, and D and PC are defined in (9) and (10), respectively.

3 Find $S_k^* \in S$ such that $F(G_{log}, S_k^*) = \max_{S_k \in S} F(G_{log}, S_k)$

4 According to the blur type of S_k^* , output the blur type of g .

Algorithm 3: for blind image restoration.

Input : degraded image y and set related parameters;
Output: image estimate;

- 1 **Step 1.** For image y , perform **Algorithm 2**;
- 2 **Step 2.** If type is Motion, the motion blur kernel is given by the support parameter estimate by Algorithm 1;
- 3 If type is out-of-focus blur, the out-of-focus blur kernel is given by the blur-SURE method;
- 4 If the type is Gaussian, the Gaussian blur kernel is given by the support parameter estimate by Algorithm 1 and the variance estimate by the blur-SURE method;
- 5 **Step 3.** Constructing blur matrix H by the identified blur type and estimated blur support parameters;
- 6 Let initial point $x^1 = y$ and let M be maximum iteration number;
- 7 **for** $k = 1:M$
 - 8 do $x^{k+1} = (H^T H + \mu I)^{-1} (H^T y + \mu x^k)$;
 - 9 Input x^{k+1} into CNN;
 - 10 Output residual image r^{k+1} ;
 - 11 $x^{k+1} := x^{k+1} - r^{k+1}$;
- 12 **end**
- 13 Output image estimate: x^{k+1}

3.4. Comparison with related works

First, there exist several methods for estimating blur support parameters. To estimate blur support size, You and Kaveh presented an iterative threshold pruning alternating minimization method [30], and Chen and Yap presented a discrete spatial method [31,32]. The two methods may estimate small support parameters of both Gaussian and uniform blur kernels. It was reported in [27] that the support parameter of Gaussian blur may be set within $[-3\sigma, 3\sigma]$ where σ is the Gaussian variance, because it contains the major energy in a Gaussian function so that a manual setting method was used to estimate blur support parameter, but this method is time consuming and its estimation accuracy very

of (3×3) dilated convolutions from first layer to the last layer are set to 1, 2, 3, 4, 3, 2 and 1, respectively. A dilated filter with dilation factor s can be simply interpreted as a sparse filter of size $(2s+1) \times (2s+1)$ where only 9 entries of fixed positions can be non-zeros. Hence, the equivalent receptive field of each layer is 3, 5, 7, 9, 7, 5 and 3. So it can obtain that the receptive field of the CNN network is 33×33 . The number of feature maps in each middle layer is set to 64.

Finally, the proposed CNN-based algorithm for blind image deconvolution is summarized in **Algorithm 3** and Fig. 4 displays its block diagram, where the architecture of the CNN is described in Fig. 5 and the blur-SURE method is given by Xue and Blur [26].

depend on the Gaussian variance. Recently, Xue and Blu proposed the blur-SURE estimation-based method [26], where the Gaussian variance is first estimated and then the support parameter is estimated approximately by using the rule of empirical identification. Oliveira et al. proposed the power-spectrum estimation method for linear motion and Out-of-Focus blur kernels [60]. This method can not be used to estimate the Gaussian blur kernel. Recently, Dash et al. presented a learning-based method for motion blur estimation [74], Wang et al. proposed an improved scheme for motion blur parameter estimation [75]. On the other side, for shake motion blur estimation, Liu et al. proposed a learning-based method [33], Xu et al. proposed a deep learning method [35], and Li et al. presented a learning method based on directional similarity and filtering [34]. This learning method can not estimate the motion blur kernel fast.

Second, for blur type identification, Liu et al. proposed a Bayes classifier using handcrafted blur features [37]. Su et al. proposed a blur detection and classification method based on the alpha channel feature [38]. Palm et al. [40] introduced a convolutional neural network for blur classification. Rugna and Konik proposed [41] a automatic segmental technique to discriminate blurry regions. Recently, Yan and Shao proposed [41] a deep learning method for blur kernel estimation. The blur type is first identified by using both the logarithmic power spectrum normalized technique and a supervised deep neural network, and then the blur support parameter is estimated by a regression neural network. In contrast, the proposed blur type identification method not only uses logarithmic power spectrum [61], but also uses both log-Gabor filter [45] and feature similarity index [44] for optimal blur type identification. To the best of our knowledge, this is the first time that joint three techniques have been applied to the problem of blur type identification. On the other side, instead of the regression neural network [44], the proposed blur support parameter estimation is based on the feature line detection technique. Because of no learning network parameters, the proposed feature line detection algorithm has fast computation performance.

Lastly, compared with current blind image restoration methods, the proposed joint method can enhance the equality of restored images under three types of unknown blur kernels since we joint a high accurate blur kernel estimation and popular CNN.

4. Experiments

In this section, we give experimental results to illustrate the effectiveness of the proposed joint blur kernel estimation and CNN-based method for blind image restoration. We evaluate the algorithm performance by using the PSNR, ISNR [55], FSIM [44] three evaluation indexes:

$$\text{PSNR} = 10\log_{10}\left(\frac{255^2}{\|\hat{x} - x\|^2/N}\right) \quad (16)$$

$$\text{ISNR} = 10\log_{10}\left(\frac{\|x - y\|^2}{\|x - \hat{x}\|^2}\right) \quad (17)$$

where x express the original image, y express the observed image, \hat{x} express reconstruction image, and N is the total pixels of the image. FSIM calculation can see from paper [44]. It is seen that larger PSNR, ISNR, FSIM values imply the better performance of image reconstruction algorithm. The simulation results are conducted in MATLAB (R2014b) platform of PC computer with 3.3GHz Intel Xeon CPU and NVIDIA Quadro K620 GPU.

4.1. Blur support parameter estimation results

This section illustrates the effectiveness of the proposed Algorithm 1.

We study benchmark image data from MSRA [73] salient object database.¹ Without the loss of generality, we randomly collect 11 image library from MSRA image library [73], which contains 2610 images where each image library consists of 235, 330, 233, 318, 245, 195, 244, 159, 246, 182, and 223 images, respectively. For our testing images, the collected images are degraded by motion, uniform, and Gaussian blurs with four different support parameters, respectively. When performing the proposed algorithm, we take the parameters $\tau = \text{ceil}(m/20)$ and $\tau = \text{ceil}(n/20)$ for Gaussian blur kernel, $\tau = \text{ceil}(m/30)$ and $\tau = \text{ceil}(n/30)$ for uniform and motion blur kernel, and $\varepsilon = \tau - 5$, respectively.

When recognizing the feature line at each row (or column) by using a positive integer number $n/2$ or $m/2$, we can obtain the best average accuracy being about 99.54%. This is because selecting other positive integer number will reduce the average accuracy. For example, if we recognize the feature line by using a positive integer number n or m , then the average accuracy is 40.96%. When $n/3$ or $m/3$ is used, the average accuracy is 98.62%.

Table 1 lists the results of blur support parameter estimation in noise-free case. We see that the proposed **Algorithm 1** can estimate the Gaussian blur support parameters at average accurate rate 99.6% above, uniform blur support parameter at average accurate rate 99.5% above, and motion blur support parameter at average accurate rate 99.3% above, respectively. **Table 2** listed the results of blur support parameter estimation in Gaussian noise with three different variances. We see that the proposed **Algorithm 1** can estimate the Gaussian blur support parameters at average accurate rate of 99.5% above, uniform blur support parameter at average accurate rate of 99.4% above, and motion blur support parameter at average accurate rate of 99% above, respectively.

For the robustness experiment, **Table 3** lists the average results obtained by the proposed **Algorithm 1**, based on independent 100 Monte Carlo simulations in Gaussian noise with variance being 10^{-3} . We observe that accuracy changes are very small for all three blur kernels. This implies that the proposed blur support parameter estimation method is robust in low-level noise.

Furthermore, **Fig. 6** displays the results of estimating four different support parameters by the proposed algorithm as each image library in noise-free case, where Gaussian blur has variance being 4. **Fig. 7** displays the results of estimating four different support parameters by the proposed algorithm as each image library in Gaussian noise with variance being 10^{-3} . From **Figs. 6** and **7** we see that accuracy changes are also small for each image library with three blur kernels. This implies that the proposed **Algorithm 1** is very effective on different images in low-level noise.

Finally, we compare the proposed **Algorithm 1** with conventional support parameter estimation methods. For the cases of both Gaussian and uniform blur kernels, we performed the minimum cyclic-shift correlation (MCSC) criterion method [31] and two discrete spatial approaches [32]: maximum average square difference estimator (MASD) and maximum average absolute difference estimator (MAAD). For the case of motion blur kernel, we performed the MCSC method, the Radial basis function neural network (RBFNN) method [74], and improved scheme [75]. We considered 128×128 Lena as testing image since MCSC, MASD, and MAAD method are time-consume. In addition, when performing the improved scheme [75], we employed $L = kN/d$ to compute the length of motion blur, where N is the width of the image, the distance between two valleys closest to the center of the image is $2d$, and k indicates the subdivision times. **Table 4** displays the computed results of blur support parameter estimation by six methods, where the Bold indicates the results by our method and the Bold and Italic indicates inaccurate results by other methods. From

¹ MSRE:<https://mmcheng.net/msra10k/>.

Table 1
Results of blur support parameter estimation in noise-free case.

PSF	Gaussian blur($\sigma = 3$)		Gaussian blur($\sigma = 4$)		Gaussian blur($\sigma = 5$)		Motion blur			Uniform blur		
	Support parameter	Correct number	Accuracy	Support parameter	Correct number	Accuracy	Support parameter	Correct number	Accuracy	Support parameter	Correct number	Accuracy
5 × 5	2609	99.96%	2608	99.92%	2609	99.96%	1 × 11	2604	99.77%	5 × 7	2606	99.85%
7 × 7	2606	99.85%	2606	99.85%	2605	99.81%	1 × 17	2599	99.58%	7 × 9	2596	99.46%
9 × 9	2605	99.81%	2606	99.85%	2606	99.85%	1 × 23	2591	99.27%	9 × 11	2597	99.50%
11 × 11	2582	98.93%	2599	99.58%	2603	99.73%	1 × 29	2570	98.47%	11 × 13	2597	99.50%

Table 2
Results of blur support parameter estimation in Gaussian noise.

Image database	PSF	Gaussian blur($\sigma = 3$)		Gaussian blur($\sigma = 4$)		Gaussian blur($\sigma = 5$)		Motion blur			Uniform blur			
		Noise variance	Support parameter	Correct number	Accuracy	Correct number	Accuracy	Correct number	Accuracy	Support parameter	Correct number	Accuracy	Support parameter	Correct number
10 ⁻⁵	5 × 5	2609	99.96%	2609	99.96%	2609	99.96%	11	2596	99.46%	5 × 7	2608	99.92%	
	7 × 7	2606	99.85%	2605	99.81%	2605	99.81%	17	2589	99.20%	7 × 9	2600	99.62%	
	9 × 9	2605	99.81%	2606	99.85%	2606	99.85%	23	2595	99.43%	9 × 11	2598	99.54%	
	11 × 11	2580	98.85%	2605	99.81%	2603	99.73%	29	2556	97.93%	11 × 13	2586	99.08%	
	10 ⁻⁴	5 × 5	2609	99.96%	2609	99.96%	2609	99.96%	11	2593	99.35%	5 × 7	2606	99.85%
	7 × 7	2606	99.85%	2605	99.81%	2605	99.81%	17	2588	99.16%	7 × 9	2601	99.66%	
10 ⁻³	9 × 9	2602	99.69%	2606	99.85%	2605	99.81%	23	2594	99.39%	9 × 11	2605	99.81%	
	11 × 11	2480	98.02%	2603	99.73%	2603	99.73%	29	2559	98.05%	11 × 13	2596	99.46%	
	5 × 5	2609	99.96%	2587	99.12%	2609	99.96%	11	2592	99.31%	5 × 7	2590	99.23%	
	7 × 7	2602	99.69%	2580	98.85%	2601	99.66%	17	2588	99.16%	7 × 9	2594	99.39%	
	9 × 9	2591	99.27%	2605	99.81%	2603	99.73%	23	2586	99.08%	9 × 11	2584	99.00%	
	11 × 11	2533	97.05%	2592	99.31%	2601	99.66%	29	2568	98.39%	11 × 13	2589	99.20%	

Image database	PSF	Gaussian blur($\sigma = 3$)		Gaussian blur($\sigma = 4$)		Gaussian blur($\sigma = 5$)		Motion blur		Uniform blur	
		Noise variance	Support parameter	Correct number	Accuracy	Correct number	Accuracy	Support parameter	Correct number	Accuracy	Support parameter
10^{-3}	5×5	2593	99.34%	2588	99.14%	2590	99.24%	11	2607	99.92%	5×7
	7×7	2601	99.64%	2600	99.62%	2601	99.64%	17	2597	98.97%	7×9
	9×9	2598	99.54%	2601	99.63%	2598	99.54%	23	2578	99.43%	9×11
	11×11	2537	97.20%	2606	99.85%	2607	99.89%	29	2587	98.20%	11×13

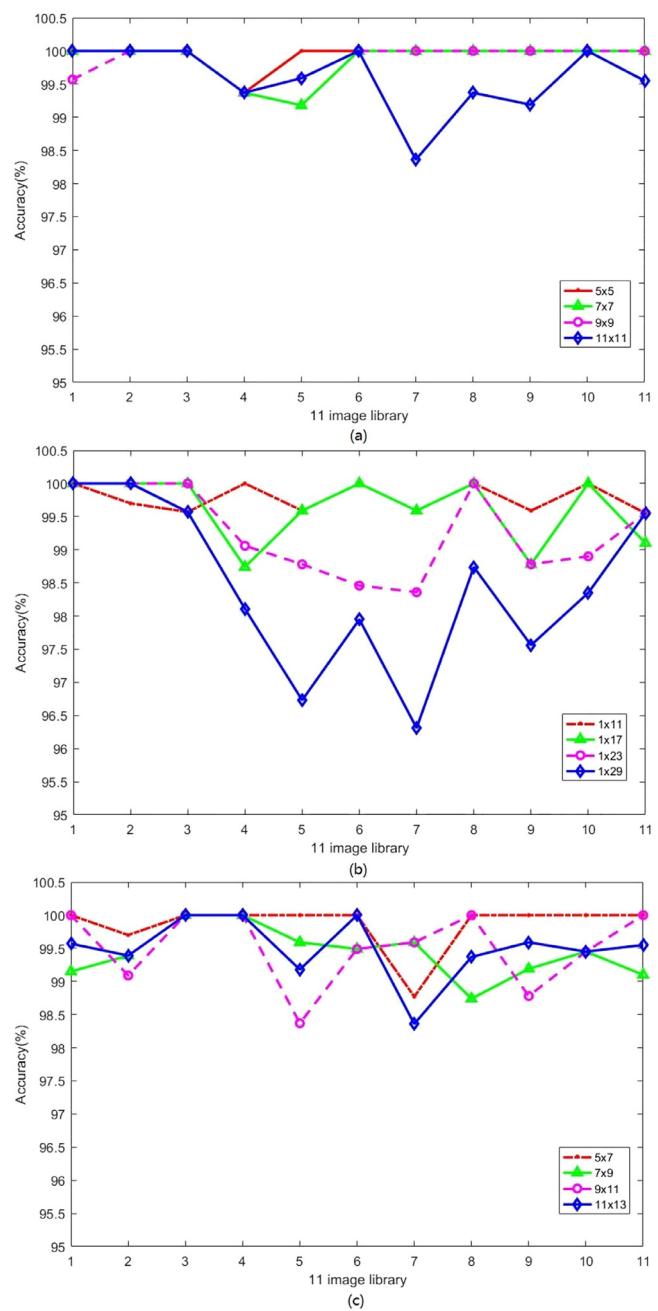


Fig. 6. Accuracy results of support parameter identification of three blurs as different image library in noise-free case. (a) Gaussian blur; (b) Motion blur; (c) Uniform blur.

Table 4 we see that the proposed **Algorithm 1** is more effective on the cases of small and large-size support parameters than other methods. Moreover, the proposed **Algorithm 1** is also faster than other methods.

4.2. Blur type identification results

This section illustrates the effectiveness of the proposed **Algorithm 2**.

We collect 2610 images which are randomly selected from MSRA image dataset [73]. For our study we generate 429 logarithmic normalized images by using the 'moto' original image and thus build a dictionary training library as below:

Table 4

Computed results of blur support parameters estimated by six methods.

PSF	Gaussian blur		Gaussian blur		Gaussian blur		Gaussian blur	
	$7 \times 7(\sigma^2 = 3.5)$		$11 \times 11(\sigma^2 = 5.5)$		$23 \times 23(\sigma^2 = 9.5)$		$27 \times 27(\sigma^2 = 10.5)$	
Methods	Estimated	Time(s)	Estimated	Time(s)	Estimated	Time(s)	Estimated	Time(s)
MCSC Criterion(2005) [31]	7×7	69.03	11×11	68.77	25×25	94.44	23×23	178.64
MASD estimator(2006) [32]	7×7	562.98	11×11	452.45	21×21	449.31	25×25	568.97
MAAD estimator(2006) [32]	7×7	419.72	11×11	506.56	21×21	403.38	25×25	501.38
Proposed method	7×7	0.25	11×11	0.55	23×23	0.34	27×27	0.45
PSF	Uniform blur		Uniform blur		Uniform blur		Uniform blur	
	7×9		9×11		19×21		23×25	
Methods	Estimated	Time(s)	Estimated	Time(s)	Estimated	Time(s)	Estimated	Time(s)
MCSC Criterion(2005) [31]	7×9	239.44	9×11	288.41	21×15	287.97	21×21	241.14
MASD estimator(2006) [32]	7×9	1125.91	9×11	988.83	25×21	809.11	27×25	807.19
MAAD estimator(2006) [32]	7×9	1013.45	9×11	790.66	25×21	816.05	27×25	809.23
Proposed method	7×9	0.39	9×11	0.41	19×21	0.66	23×25	0.62
PSF	Motion blur		Motion blur		Motion blur		Motion blur	
	1×7		1×11		1×23		1×27	
Methods	Estimated	Time(s)	Estimated	Time(s)	Estimated	Time(s)	Estimated	Time(s)
MCSC Criterion(2005) [31]	1×7	68.19	1×15	67.86	1×23	94.39	1×31	132.36
RBFNN method(2014) [74]	1×8.1	1.91	1×11.8	2.01	1×23.8	2.25	1×28.9	2.59
Improved scheme(2017) [75]	1×7.26	0.47	1×11.15	0.41	1×23.52	0.44	1×27.53	0.39
Proposed method	1×7	0.28	1×11	0.27	1×23	0.27	1×27	0.39

Table 5

Average results of blur type identification in Gaussian noise.

Image library	Noise variance	10^{-3}		10^{-2}	
		N_c	CR	N_c	CR
Gaussian	7×7	2610	100%	2610	100%
	9×9	2610	100%	2610	100%
	11×11	2609	99.96%	2584	99.00%
Motion	$\omega = 0^\circ$	2610	100%	2591	99.27%
	$\omega = 30^\circ$	2610	100%	2593	99.35%
	$\omega = 60^\circ$	2610	100%	2591	99.27%
Out-of-focus	$R = 5$	2610	100%	2588	99.16%
	$R = 7$	2610	100%	2578	98.77%
	$R = 9$	2561	98.12%	2523	96.67%

- (1) Gaussian blur: variance range taken as [1.0: 0.5: 31.0] and blur size range taken as [3,31], where variance value is small than blur size. The total number of images is 230.
- (2) Motion blur: length range taken as [5: 2: 69], angle range taken as [0,30:10:180], where the angle is 0° if the length taken is large than 23. The total number of images is 177.
- (3) Out-of-focus blur: radius taken as [2: 1: 23]. The total number of images is 22.

We evaluate the performance of blur type identification based on classification rate [41]:

$$CR = 100 \frac{N_a}{N_t} (\%) \quad (18)$$

where N_a is the number of correct classified images, and N_t is the total number of images.

Table 5 lists the average results obtained by the proposed **Algorithm 2**, based on independent 100 Monte Carlo simulations in Gaussian noise with variance being 10^{-3} . From **Table 5** we see that the proposed **Algorithm 2** can obtain an accurate rate of 99.42%. Moreover, we observe that accuracy changes are very small for all three blur kernels. This implies that the proposed **Algorithm 2** is robust in low-level noise.

Furthermore, we compare the proposed **Algorithm 2** with six blur type identification algorithms: two existing blur type identification methods based on handcrafted features [37,38], the back-propagation neural network (NN) [39], the convolutional neural

Table 6

Comparison of blur kernel type identification by seven algorithms in Gaussian noise.

Method	CR1 (200)	CR2 (500)
[37] (2008)	78.1%	79.4%
[38] (2011)	80.7%	81.5%
[42] (2005)	76.9%	78.8%
NN [39] (1997)	89.7%	90.2%
CNN [40] (2012)	92.2%	93.9%
Deep Learning [41] (2016)	94.5%	95.2%
Proposed	96.5%	97.6%

network (CNN) [40], the support vector machine(SVM) method [42], and the deep learning method [41]. We use testing datasets: (1) Berkeley segmentation public dataset,² including 200 images used in denoising algorithms [62,63] and image quality assessment [64], we called this database as CR1, and (2) Pascal VOC 2007 dataset,³ including 4952 images used in [65], where we randomly choose 500 images,we called this database as CR2. For our experiments, we randomly select a blur kernel for degradation and its blur parameters are set in the case: 1) Gaussian blur: $\sigma = [1 : 0.5 : 5]$; 2) Motion blur: $\omega = [30 : 10 : 180]$; 3) Out-of-focus blur: $R = [2 : 1 : 23]$. **Table 6** lists their computed results. From **Table 6** we see that the proposed **Algorithm 2** gives the best results. This is because the proposed **Algorithm 2** combines the logarithmic power spectrum technique [61], the log-Gabor filter [45], and the feature similarity index [44] together.

4.3. Blind image restoration experiments

This section illustrates the effectiveness of the proposed **Algorithm 3** for blind image restoration.

For our test, we study blurred images generated by three blur kernels (Gaussian, Motion, Out-of-focus) and Gaussian noise with variance being 10^{-3} was further added to the blurred images to form the degraded images. We compare the proposed

² Berkeley:<https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/resources.html>.

³ Pascal:<http://host.robots.ox.ac.uk/pascal/VOC/voc2007/index.html>.

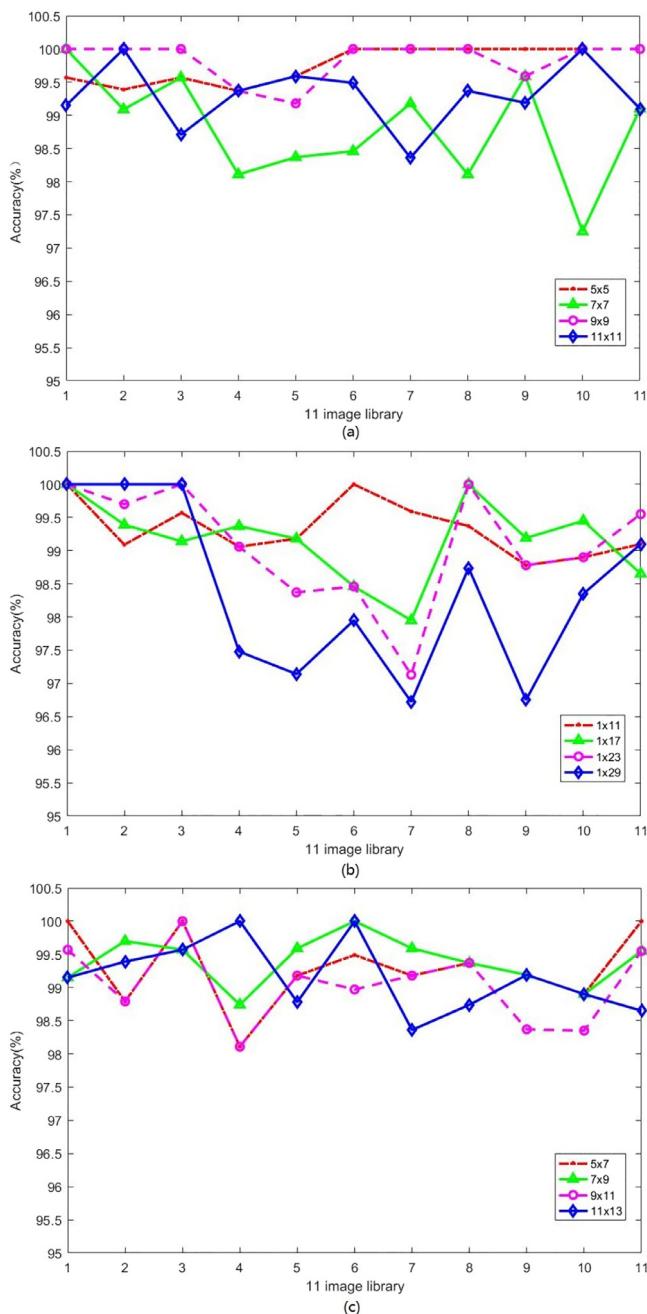


Fig. 7. Accuracy results of support parameter identification of three blurs as different image library in Gaussian noise case. (a) Gaussian blur; (b) Motion blur; (c) Uniform blur.

Algorithm 3 with state-of-art blind image restoration methods: the normalized sparsity measure (NSM)⁴ [66], the alternating maximum a posteriori estimation method (MAP)⁵ [67], the internal patch recurrence method (IPR)⁶ [68], the variational dirichlet estimation method (VD)⁷ [69], the blind image deblurring using dark channel prior (DCP)⁸ [24], and the neural approach to blind motion deblurring (NBMD)⁹ [51].

⁴ NSM:http://www.cs.nyu.edu/~dilip.wordpress/?page_id=159.

⁵ MAP:<http://zoi.utia.cas.cz/download>.

⁶ IPR:<http://www.wisdom.weizmann.ac.il/~vision/BlindDeblur.html>.

⁷ VD:<http://decsai.ugr.es/vip/resources/VDBKE.html>.

⁸ DCP:<http://vllab1.ucmerced.edu/~jinshan/projects/dark-channel-deblur/>.

⁹ NBMD:<https://projects/ayanc.org/nndeblur/>.

To perform the proposed **Algorithm 3**, the CNN consists of seven layers with three different blocks. Similar to dataset used in [9],¹⁰ we collect a large dataset which includes 400 Berkeley segmentation dataset images, 400 images selected from validation set of ImageNet database and 4744 images of Waterloo Exploration Database. We crop these images into small patches of size 35×35 and select $N = 256 \times 4000$ patches for our training.

First, we consider 5×5 , 7×7 , and 9×9 Gaussian blur kernels with standard deviation being 2.5, 3.5 and 5.5, respectively. Restored results by six different methods are shown in Figs. 8, 9 and 10 respectively. We observe that the restored images by the proposed **Algorithm 3** are visually much better than other algorithms. For example, as shown in the red box shown in the Fig. 10, we see that recovered details are much clearer than other algorithms.

Second, we consider 1×11 , 1×17 , and 1×21 motion blur kernels with angle 0° , respectively. Restored results by six methods are shown in Figs. 11, 12 and 13 respectively. From the Figs. 11–13 we see that the NBMD method is not suitable for dealing with blurred color images. The restored images by the proposed **Algorithm 3** are visually better than other algorithms. For example, as shown in the red box shown in the Fig. 12, we see that recovered details are much clearer than the VD method and DCP method.

Third, we consider $R = 5, 8, 10$ out-of-focus blur kernels, respectively. Restored results by six methods are shown in Figs. 14–16, respectively. From the Figs. 14–16 we see that the restored images by the proposed **Algorithm 3** are visually better than other algorithms. For example, as shown in the red box shown in the Fig. 15, we see that image details recovered by the proposed **Algorithm 3** are much clearer than other algorithms.

Furthermore, Table 7 listed their computed results of ISNR, PSNR, SSIM, and running time. Table 7 shown in the bold font shows that the proposed **Algorithm 3** gives the best index values, compared to other algorithms. Moreover, the proposed algorithm required the shortest computation time.

To show the effectiveness of the proposed **Algorithm 3** on the case of large blur kernel size, we consider 27×27 Gaussian blur with standard deviation being 11, 1×27 motion blur, and $R = 30$ out-of-focus blur, respectively. Figs. 17–19 display the restored images by six algorithms. It is seen that the restored images by the proposed **Algorithm 3** are visually better than other algorithms. Furthermore, Table 8 listed their computed results on ISNR, PSNR, SSIM and running time. From Table 8 we see that the proposed **Algorithm 3** is also superior to other algorithms.

To illustrate the robustness of the proposed **Algorithm 3**, we study two well-known image datasets: Berkeley public dataset and Pascal VOC 2007 dataset. Similar to [41], we randomly chooses 100 images and three blur kernels for degradation. Their blur parameters are set in the case: Gaussian blur: $\sigma = [1 : 0.5 : 5]$ or Motion blur: $L = [5 : 2 : 27]$ or out-of-focus blur: $R = [2 : 1 : 27]$. We compare the proposed **Algorithm 3** with the deep learning algorithm [41]. Table 9 lists the average results obtained by the proposed **Algorithm 3** over 100 runs, in terms of PSNR, SSIM, GMSD [71] and GS [72]. It is seen that the proposed **Algorithm 3** gives the bigger values of PSNR, SSIM, and GS, and gives the smaller values of GMSD. Therefore, the proposed **Algorithm 3** is superior to the deep learning algorithm.

To illustrate the effectiveness of the CNN in the proposed joint method, we also performed three non-blind image restoration methods based on the blur kernel estimate by Algorithms 1 and 2. The tree non-blind image restoration methods are Multi-Wiener SURE-LET method (MWSURE)¹¹ [3], Joint statistical modeling

¹⁰ Deep CNN Denoise for IR:<http://github.com/cszn/ircnn>.

¹¹ SURE:<http://www.ee.cuhk.edu.hk/~tblu/msonsite/phps/demos.php>.

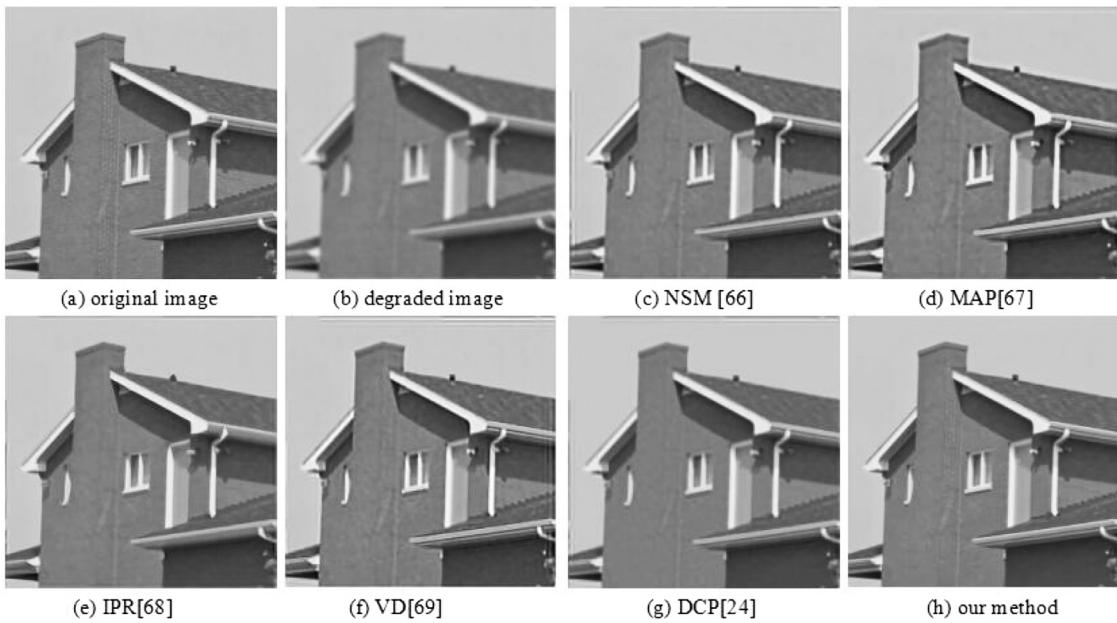


Fig. 8. Restored results of blurred image with 5×5 Gaussian blur and variance being 2.5 by six methods.

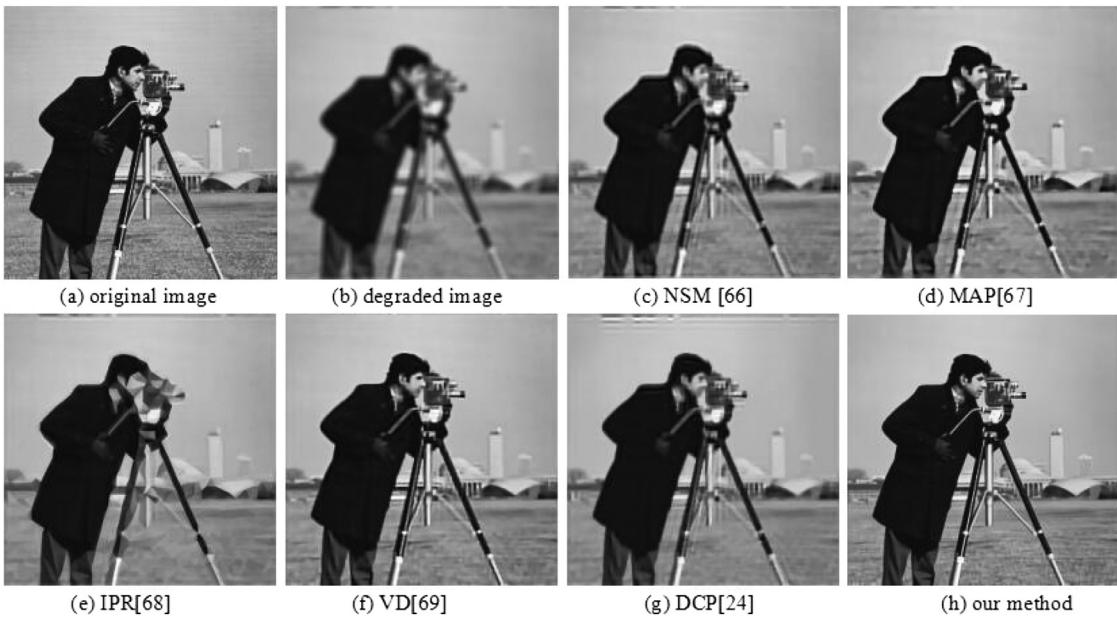


Fig. 9. Restored results of blurred image with 7×7 Gaussian blur and variance being 3.5 by six methods.

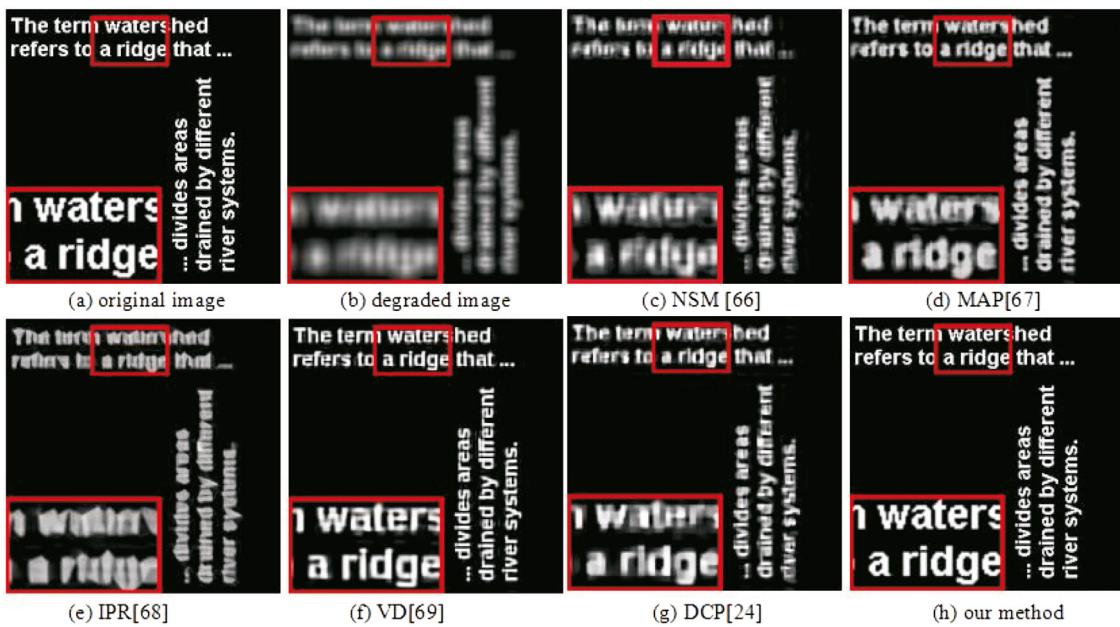


Fig. 10. Comparison of blurred image with 9×9 Gaussian blur and variance being 5.5 by six methods.

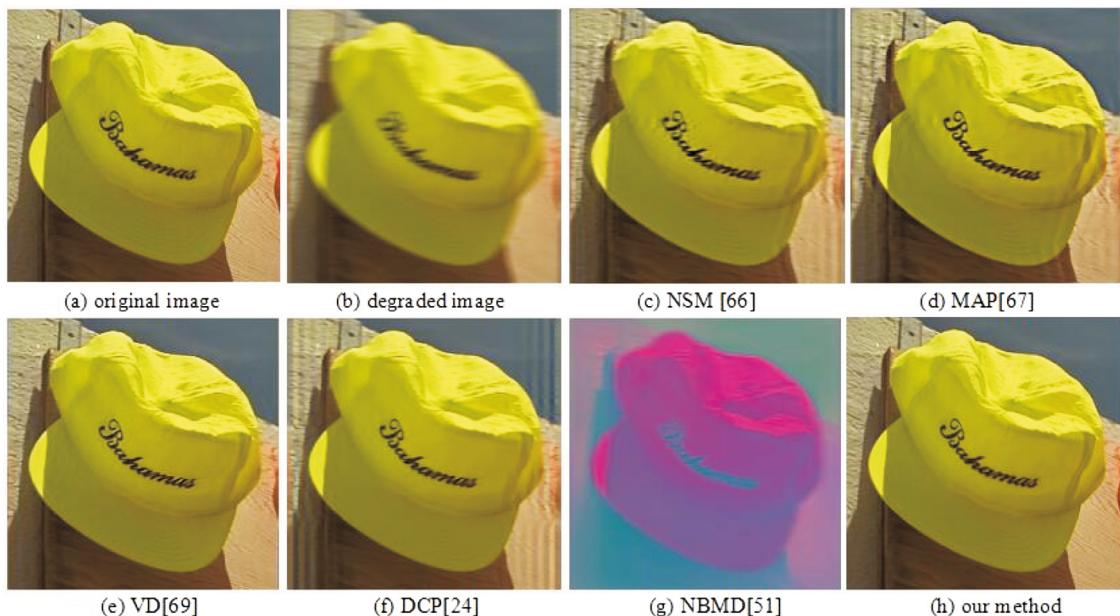


Fig. 11. Restored results of blurred image with 1×11 Motion blur by six methods.

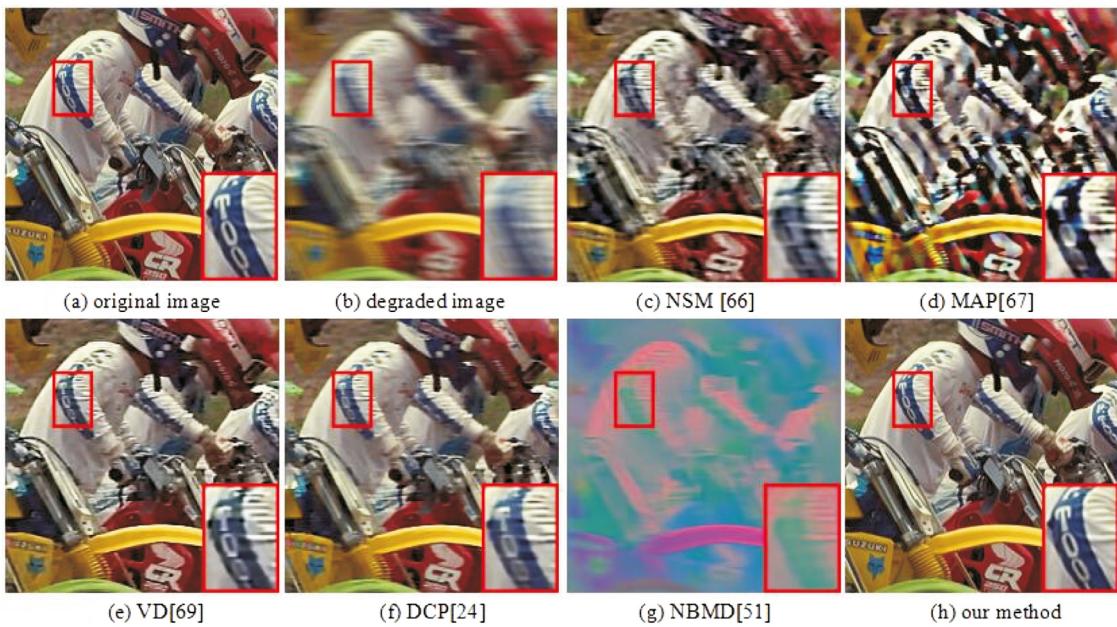


Fig. 12. Restored results of blurred image with 1×17 Motion blur by six methods.



Fig. 13. Restored results of blurred image with 1×21 Motion blur by six methods.

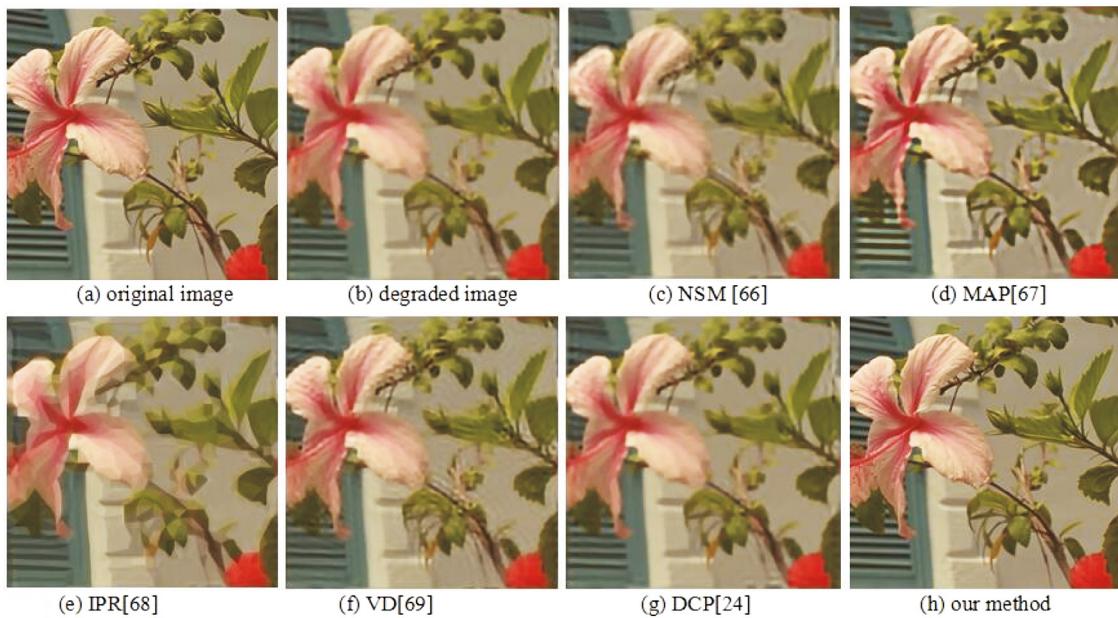


Fig. 14. Restored results of blurred image with out-of-focus blur ($R = 5$) by six methods.

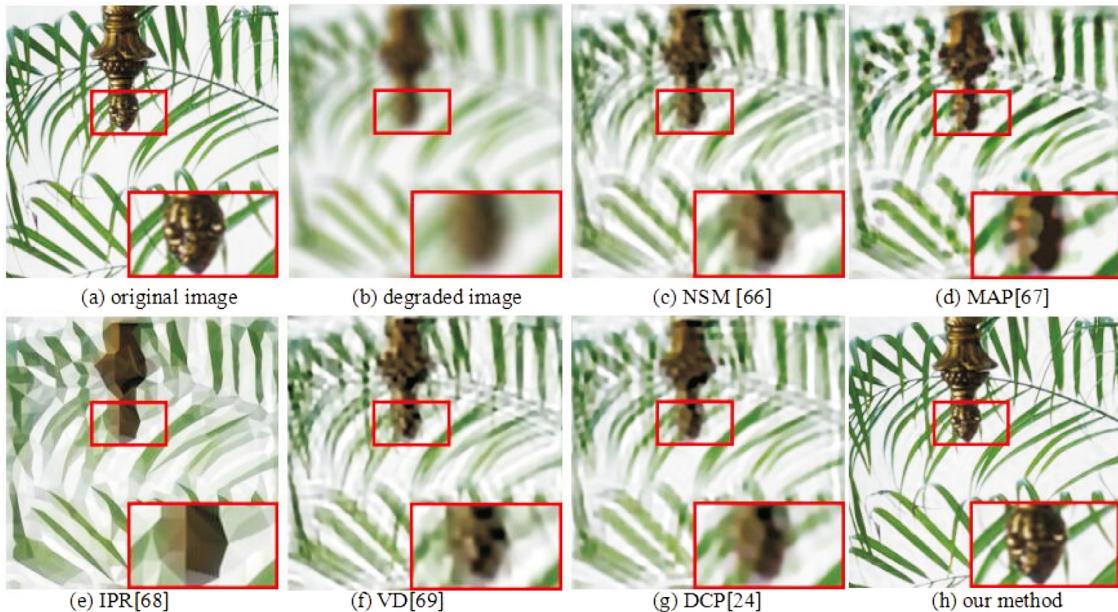


Fig. 15. Restored results of blurred image with out-of-focus blur ($R = 8$) by six methods.

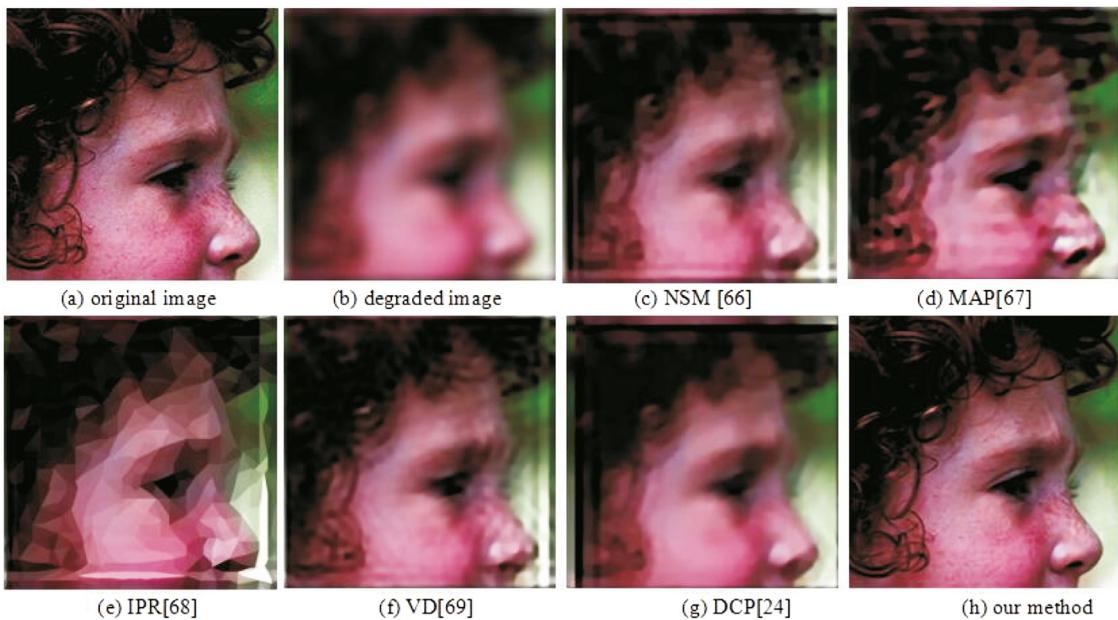


Fig. 16. Restored results of blurred image with out-of-focus blur ($R = 10$) by six methods.

Table 7

Computed results of deblurred images in three blur kernels.

Gaussian blur												
Evaluation criteria	ISNR			SSIM			PSNR			Time		
Image	House	Camera	Letter	House	Camera	Letter	House	Camera	Letter	House	Camera	Letter
NSM [66] (2011)	0.115	1.611	0.074	0.829	0.786	0.806	28.363	23.634	14.214	53.852	55.334	79.701
MAP [67] (2013)	-1.284	4.447	-0.746	0.824	0.853	0.817	26.963	26.470	13.393	52.510	56.285	56.285
IPR [68] (2014)	1.562	0.276	0.143	0.990	0.974	0.984	29.810	22.299	14.282	135.003	282.782	355.667
VD [69] (2015)	0.414	5.253	5.984	0.862	0.885	0.915	28.662	27.275	20.123	17.753	10.327	15.241
DCP [24] (2016)	2.095	2.699	3.450	0.989	0.982	0.984	30.343	24.722	17.589	83.305	77.673	110.371
Our	11.466	11.448	19.259	0.975	0.960	0.987	39.714	33.471	33.398	5.023	5.195	4.992

Linear motion blur												
Evaluation criteria	ISNR			SSIM			PSNR			Time		
Image	Hat	Moto	Lena	Hat	Moto	Lena	Hat	Moto	Lena	Hat	Moto	Lena
NSM [66] (2011)	3.630	1.325	0.007	0.888	0.753	0.690	29.257	19.076	17.523	103.273	185.173	325.808
MAP [67] (2013)	0.184	-3.174	-1.402	0.845	0.658	0.652	25.812	14.577	16.114	53.477	51.262	54.070
VD [69] (2015)	6.759	5.184	7.616	0.942	0.865	0.889	32.387	22.934	25.131	34.617	36.208	29.781
DCP [24] (2016)	2.040	4.262	5.710	0.982	0.988	0.978	27.673	22.012	23.226	174.222	186.328	220.336
NBMD [51] (2016)	-0.278	-0.026	0.914	0.962	0.925	0.930	26.677	19.046	19.753	133.381	123.693	115.066
Our	20.503	24.010	23.739	0.995	0.994	0.991	46.131	41.761	41.254	4.493	4.493	4.493

Out-of-focus blur												
Evaluation criteria	ISNR			SSIM			PSNR			Time		
Image	Flower	Leaves	Girl	Flower	Leaves	Girl	Flower	Leaves	Girl	Flower	Leaves	Girl
NSM [66] (2011)	1.354	1.487	-0.639	0.753	0.609	0.697	23.715	16.382	22.073	87.797	217.590	468.175
MAP [67] (2013)	1.633	-0.143	-0.523	0.787	0.593	0.690	23.993	14.752	22.189	54.101	52.557	61.932
IPR [68] (2014)	-1.549	-2.391	0.989	0.956	0.926	0.949	20.810	20.324	15.887	355.713	590.105	525.287
VD [69] (2015)	2.797	1.590	-0.409	0.813	0.641	0.735	25.157	16.485	22.302	13.853	14.680	22.183
DCP [24] (2016)	3.098	0.570	-0.755	0.976	0.934	0.909	25.457	15.468	21.960	164.019	209.962	204.892
Our	16.262	18.731	10.014	0.987	0.963	0.923	38.623	33.626	32.726	4.493	4.462	4.711

method (JSM)¹² [6], and adaptive sparse domain selection method (ASDS)¹³ [70], respectively. We consider 11×11 Gaussian blur with standard deviation being 6.5, 1×13 motion blur, and $R = 15$ de-focus blur, respectively. Figs. 20–22 display restored results by four methods, respectively. From Figs. 20–22, we see that the proposed algorithm and MWSURE method has better restored

results than JSM and ASDS methods, but Fig. 21(c) shows that MWSURE method can't handle the case of motion blur. Furthermore, Table 10 lists their computed results. We see that the proposed Algorithm 3 can give fast the best index values of image quality than other methods.

Finally, we study two degraded images: automobile tire image taken by a camera and “Sailboat” image taken in [41], as shown in Figs. 23(a) and 24(a), respectively. Fig. 23(b)–(f) display the restored results of blurred automobile tire image by five methods

¹² JSM: <http://idm.pku.edu.cn/staff/zhangjian/IRJSM/>.

¹³ ASDS: http://www4.comp.polyu.edu.hk/~cslzhang/ASDS_AReg.htm.

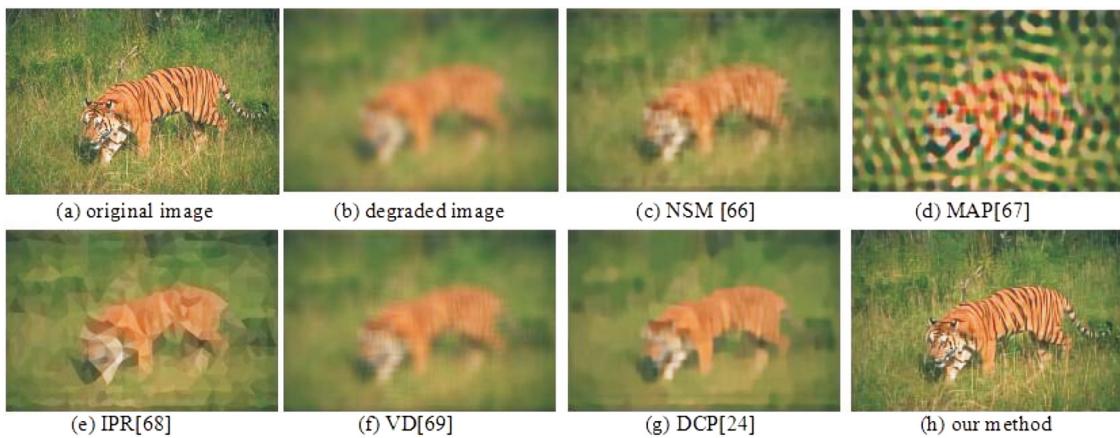


Fig. 17. Restored results of blurred image on 27×27 Gaussian blur with variance being 11 by six methods.

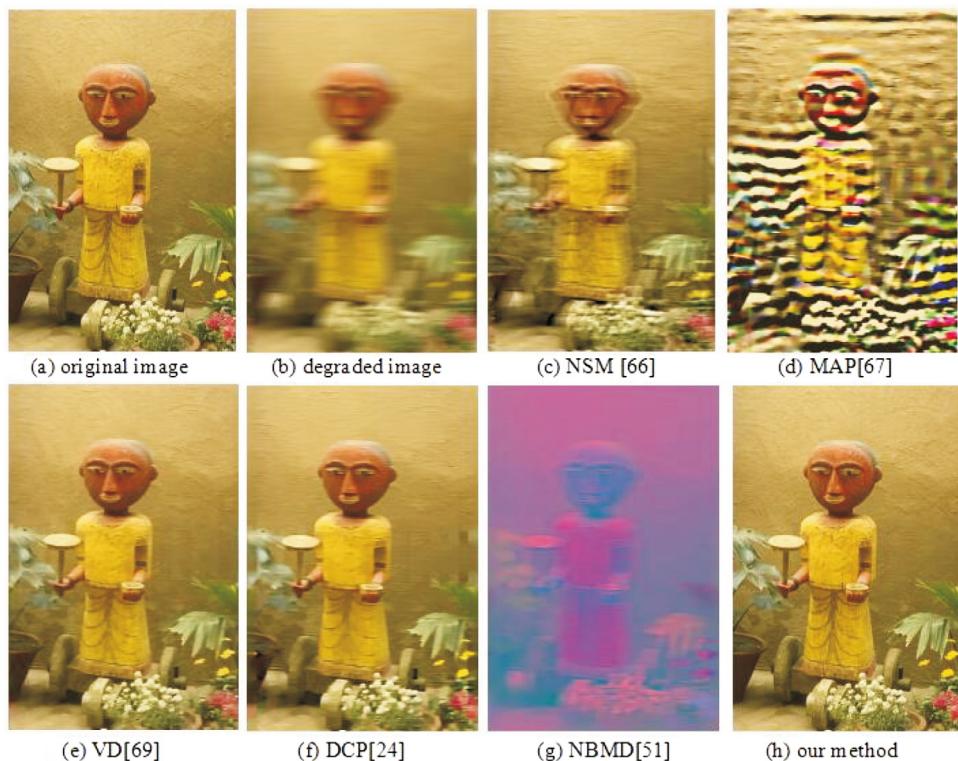


Fig. 18. Restored results of blurred image on 1×27 Motion blur by six methods.

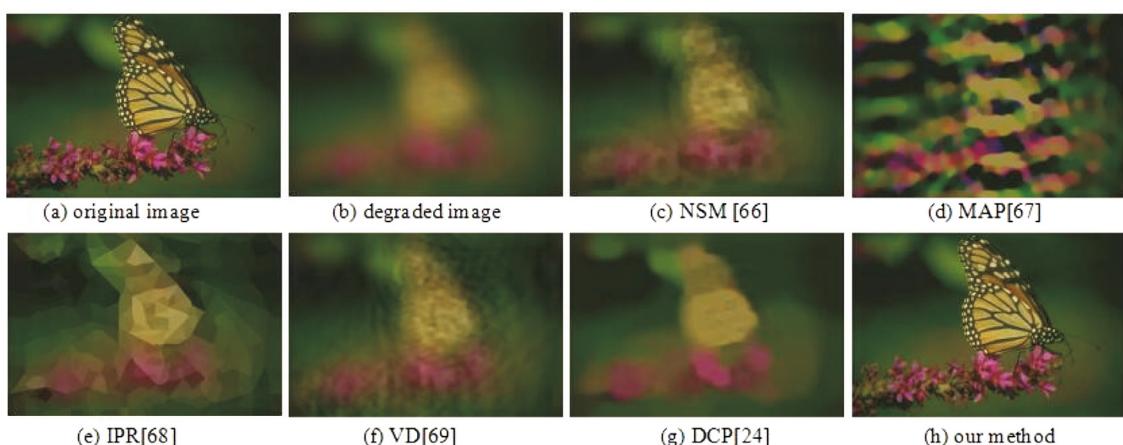


Fig. 19. Restored results of blurred image on out-of-focus blur ($R = 30$) by six methods.

Table 8

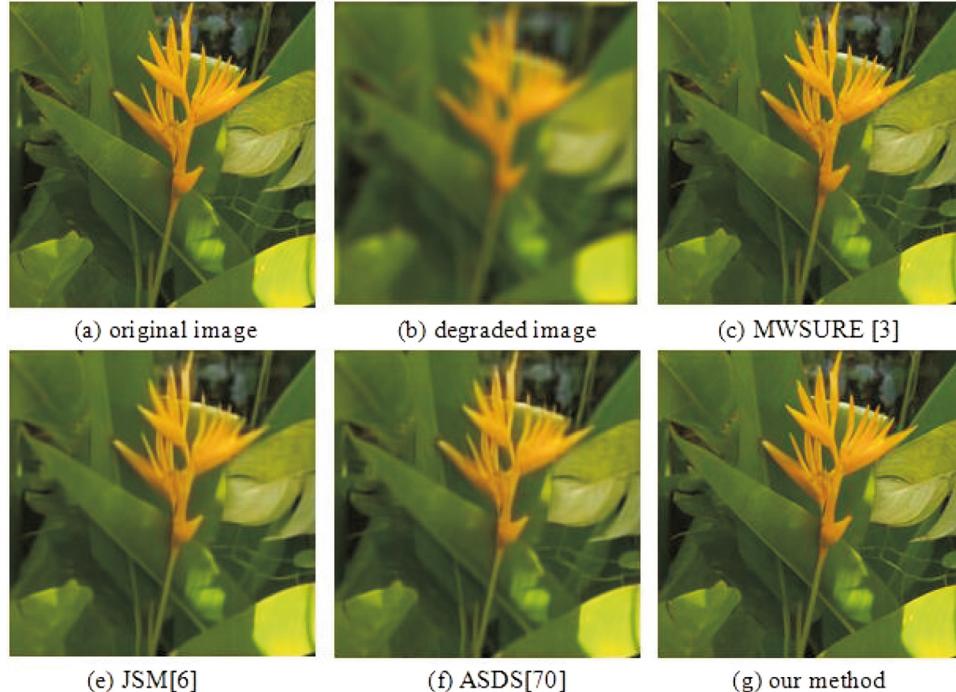
Computed results of deblurred images in the case of large-size blur kernels.

Blur kernel	Gaussian(27×27 , 11)				Out-of-focus($R = 30$)				Blur kernel	Motion(1×27)			
Image	Tiger				Butterfly				Image	Caton			
Method	ISNR	SSIM	PSNR	Time	ISNR	SSIM	PSNR	Time	Method	ISNR	SSIM	PSNR	Time
NSM [66]	0.103	0.841	19.763	485.928	-0.056	0.912	22.253	3601.143	NSM [66]	-1.224	0.940	21.509	401.063
MAP [67]	-2.978	0.740	16.682	94.817	-4.261	0.865	18.048	109.123	MAP [67]	-8.986	0.877	13.748	243.424
IPR [68]	-0.040	0.879	19.620	1958.593	-0.093	0.907	22.216	13964.320	VD [69]	6.378	0.985	29.111	161.570
VD [69]	0.008	0.844	19.669	166.562	-0.002	0.882	22.307	178.481	DCP [24]	4.004	0.976	26.737	444.260
DCP [24]	0.371	0.823	20.031	411.609	0.373	0.881	22.682	536.051	NBMD [51]	-0.564	0.909	23.491	482.464
Our	4.596	0.976	24.256	11.716	13.637	0.993	35.946	7.597	Our	22.460	1.000	45.193	7.800

Table 9

Comparison of average results based on 100 runs by two algorithms.

Method	Gaussian blur				Motion blur				Defocus blur			
	PSNR	SSIM	GMSD	GS	PSNR	SSIM	GMSD	GS	PSNR	SSIM	GMSD	GS
Input	25.11	0.6624	0.3446	0.6632	24.72	0.6413	0.3588	0.6417	22.33	0.6157	0.3850	0.6155
Deep Learning [41]	28.96	0.8786	0.1252	0.8790	27.94	0.8415	0.1417	0.8593	27.67	0.8190	0.1702	0.8267
our method	29.53	0.9361	0.0294	0.9929	30.88	0.9378	0.0473	0.9864	28.30	0.8838	0.0611	0.9740

Fig. 20. Restored results of blurred image in the case of 11×11 Gaussian blur with variance being 6.5.

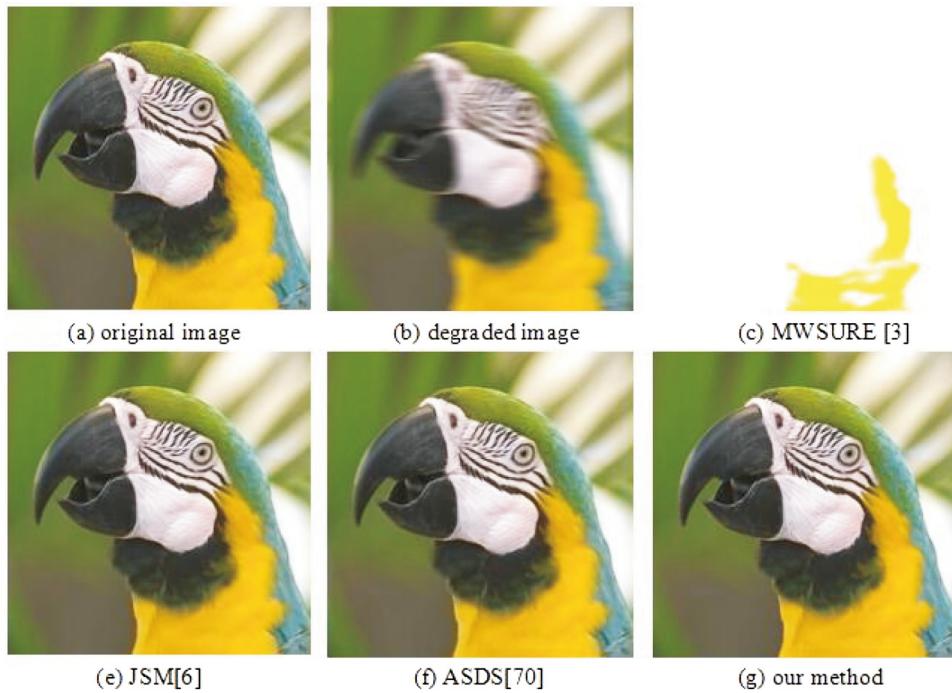


Fig. 21. Restored results of blurred image in the case of 1×13 motion blur.

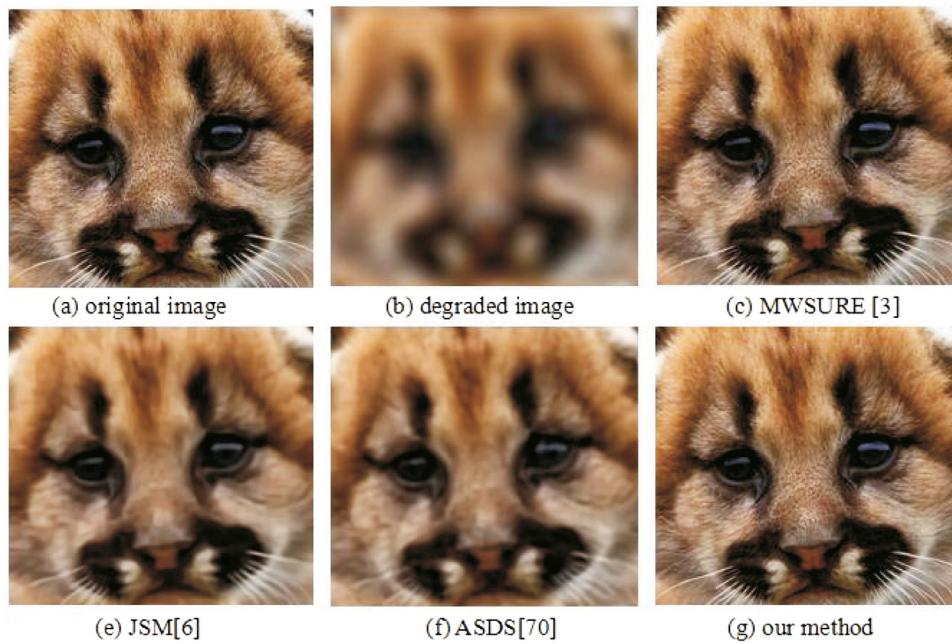


Fig. 22. Restored results of blurred image in the case of out-of-focus blur with $R = 15$.

Table 10

Comparison of computed results based on four non-blind methods.

Blur kernel	Gaussian(11×11 , 6.5)				Motion(1×13)				Out-of-focus ($R = 15$)			
Image	Plants				Parrots				Raccoon			
Evaluation criteria	ISNR	SSIM	PSNR	Time	ISNR	SSIM	PSNR	Time	ISNR	SSIM	PSNR	Time
MWSURE [3] (2013)	11.913	0.964	36.315	2.496	-17.595	0.595	5.398	2.621	11.950	0.949	32.614	4.009
JSM [6] (2014)	7.905	0.915	32.308	171.507	16.743	0.982	39.736	171.445	5.539	0.755	26.202	259.055
ASDS [70] (2011)	6.833	0.901	31.236	128.217	15.076	0.982	38.067	131.041	4.735	0.749	25.399	192.770
Our	15.397	0.979	39.800	5.179	21.961	0.993	44.955	5.538	13.374	0.958	34.038	6.006

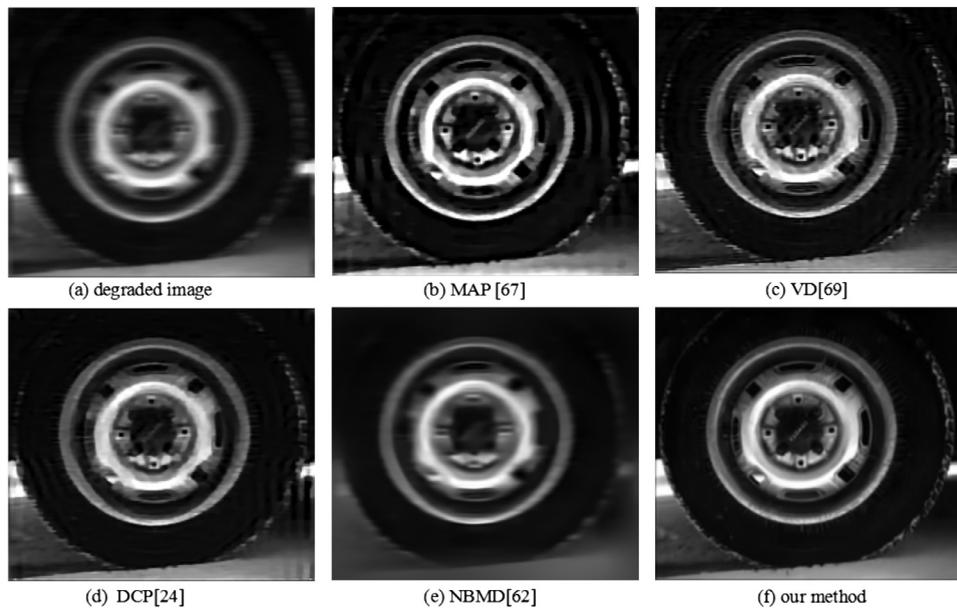


Fig. 23. Restored results of blurred automobile tire image by five methods.

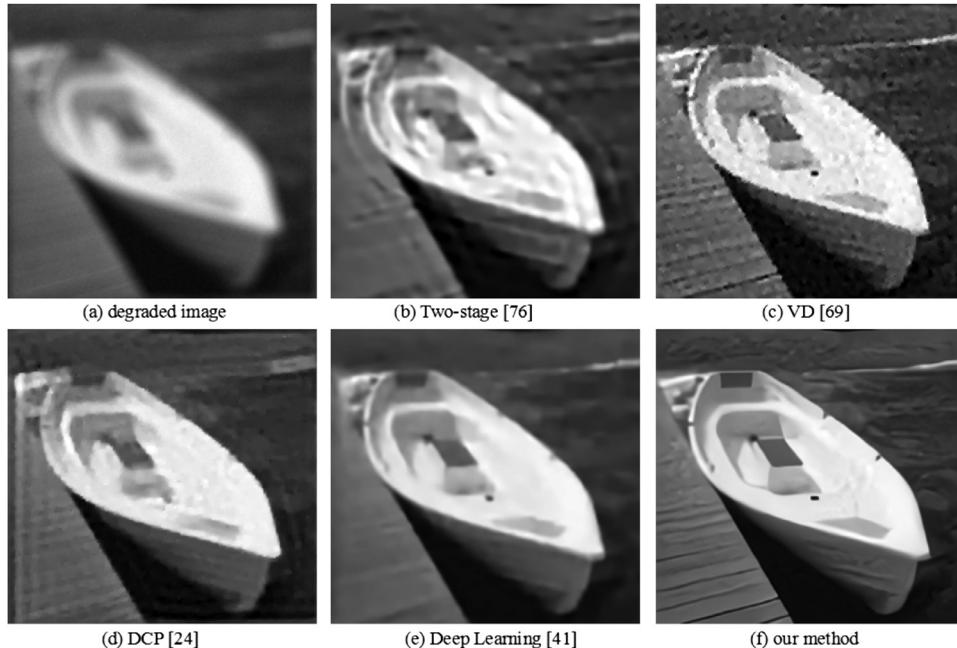


Fig. 24. Restored results of blurred "Sailboat" image by five methods.

and Fig. 24(b)–(f) display the restored results of blurred "Sailboat" image by five methods, respectively. It is seen that our restored images have clearer details than others.

5. Conclusion

This paper has proposed a joint method for blind image restoration, based on blur kernel estimation and CNN. First, the blur kernel estimation algorithm includes an automatic feature line detection algorithm for blur support parameter estimation and a dictionary learning algorithm. The proposed blur kernel estimation algorithm is used for Gaussian blur, linear motion blur, and out-of-focus blur. Second, an effective CNN is used for iterative non-blind

deconvolution, which is able to automatically learn effective image priors. The experimental result shows that the proposed joint blur kernel estimation and CNN algorithm is superior to conventional blind image restoration algorithms in terms of restoration quality and computation time. It is of significant and challenging issue to theoretically analyze the superiority of the proposed support parameter estimation algorithm. Also, it is of significant issue to study the effectiveness of the proposed method on camera-shake motion blur. These will be our further research in future.

Conflict of interest

None.

Acknowledgement

This work is supported by the National Science Foundation of China under Grant No. 61473330.

References

- [1] R.C. Gonzalez, R.E. Woods, *Digital Image Processing*, 3rd, Prentice-Hall, 2008.
- [2] D. Kundur, D. Hatzinakos, Blind image deconvolution, *IEEE Signal Process. Mag.* 13 (3) (1996) 43–64.
- [3] M.R. Azimi-Sadjadi, Strip kalman filtering for image restoration: new results, *IEEE Int. Symp. Circuits Syst.* 1 (1999) 553–556.
- [4] W. Dong, L. Zhang, G. Shi, X. Li, Nonlocally centralized sparse representation for image restoration, *IEEE Trans. Image Process.* 22 (4) (2013) 1620–1630.
- [5] J. Zhang, D. Zhao, R. Xiong, et al., Image restoration using joint statistical modeling in a space-transform domain, *IEEE Trans. Circuits Syst. Video Technol.* 24 (6) (2014) 915–928.
- [6] Y.S. Xia, H. Leung, M.S. Kamel, A discrete-time learning algorithm for image restoration using a novel l-2-norm noise constrained estimation, *Neurocomputing* 198 (2016) 155–170.
- [7] C.J. Schuler, H.C. Burger, S. Harmeling, et al., A machine learning approach for non-blind image deconvolution, in: *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1067–1074.
- [8] L. Xu, J. Ren, C. Liu, et al., Deep convolutional neural network for image deconvolution, *Adv. Neural Inf. Process. Syst.* 27 (2014).
- [9] K. Zhang, W. Zuo, S. Gu, et al., Learning deep CNN denoiser prior for image restoration, in: *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
- [10] C. Dong, C.C. Loy, K. He, et al., Image super-resolution using deep convolutional networks, *IEEE Trans. Pattern Anal. Mach. Intell.* 38 (2) (2016) 295–307.
- [11] J. Kim, J.K. Lee, K.M. Lee, Accurate image super-resolution using very deep convolutional networks, in: *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1646–1654.
- [12] C. Dong, C.C. Loy, K. He, et al., Image super-resolution using deep convolutional networks, *IEEE Trans. Pattern Anal. Mach. Intell.* 38 (2) (2016) 295–307.
- [13] S. Li, R. Fan, G. Lei, et al., A two-channel convolutional neural network for image super-resolution, *Neurocomputing* 275 (2018) 267–277.
- [14] H. Chen, X. He, C. Ren, et al., CISRDCNN: super-resolution of compressed images using deep convolutional neural networks, *Neurocomputing* 285 (2018) 204–219.
- [15] A. Kappeler, S. Yoo, Q. Dai, et al., Video super-resolution with convolutional neural networks, *IEEE Trans. Comput. Imaging* 2 (2) (2016) 109–122.
- [16] J. Zhang, J. Pan, W.S. Lai, et al., Learning fully convolutional networks for iterative non-blind deconvolution, in: *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6969–6977.
- [17] D. Krishnan, R. Fergus, Fast image deconvolution using hyper-laplacian priors, *Neural Inf. Process. Syst.* 22 (2009) 1–9.
- [18] A. Beck, M. Teboulle, Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems, *IEEE Trans. Image Process.* 18 (11) (2009) 2419–2434.
- [19] L. Sun, S. Cho, J. Wang, et al., Good image priors for non-blind deconvolution—generic vs. specific, *European Conference on Computer Vision*, Springer, 2014, pp. 231–246.
- [20] S. Siddhichai, J.A. Chambers, Wavelet transform-based noise reduction schemes to improve the noise sensitivity of the NAS-RIF algorithm for blind image deconvolution, in: *IEEE International Congress on Signal Processing*, 2000.
- [21] D. Zoran, Y. Weiss, From learning models of natural image patches to whole image restoration, in: *IEEE International Conference on Computer Vision*, 2011, pp. 479–486.
- [22] F. Chen, Y. Jiao, L. Lin, et al., Image deblurring via combined total variation and framelet, *Circuits Syst. Signal Process.* 33 (6) (2014) 1899–1916.
- [23] L. Yi, X. Lu, J. Wang, et al., Image restoration based on bregman iterative double regularization, *J. Image Graph.* 16 (3) (2011) 350–356.
- [24] J. Pan, D. Sun, H. Pfister, M.H. Yang, Blind image deblurring using dark channel prior, in: *Computer Vision and Pattern Recognition*, 2016, pp. 1628–1636.
- [25] Y. Yan, W. Ren, Y. Guo, et al., Image deblurring via extreme channels prior, in: *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6978–6986.
- [26] F. Xue, T. Blur, A novel SURE-based criterion for parametric PSF estimation, *IEEE Trans. Image Process.* 24 (2) (2015) 595–607.
- [27] F. Chen, J. Ma, An empirical identification method of gaussian blur parameter for image deblurring, *IEEE Trans. Signal Process.* 57 (7) (2009) 2467–2478.
- [28] M. Barlaud, Blind restoration of noisy blurred image using a constrained maximum likelihood method, *Opt. Eng.* 30 (4) (1991) 431–437.
- [29] Y.S. Xia, Z. Deng, W. Zheng, Analysis and application of a novel fast algorithm for 2-d ARMA model parameter estimation, *Automatica* 49 (10) (2013) 3056–3064.
- [30] Y.L. You, M. Kaveh, A regularization approach to joint blur identification and image restoration, *IEEE Trans. Image Process.* 5 (3) (1996) 416–428.
- [31] L. Chen, K. Yap, A soft double regularization approach to parametric blind image deconvolution, *IEEE Inf. Image Process.* 14 (5) (2005) 624–633.
- [32] L. Chen, K. Yap, Efficient discrete spatial techniques for blur support identification in blind image deconvolution, *IEEE Trans. Signal Process.* 54 (4) (2006) 1557–1562.
- [33] S. Liu, H. Wang, J. Wang, et al., Blur-kernel bound estimation from pyramid statistics, *IEEE Trans. Circuits Syst. Video Technol.* 26 (5) (2016) 1012–1016.
- [34] W. Li, D. Chen, Z. Lv, et al., Learn to model blurry motion via directional similarity and filtering, *Pattern Recognit.* 75 (2018) 327–338.
- [35] X. Xu, J. Pan, Y.J. Zhang, et al., Motion blur kernel estimation via deep learning, *IEEE Trans. Image Process.* 27 (1) (2018) 194–205.
- [36] L. Sun, S. Cho, J. Wang, J. Hays, Edge-based blur kernel estimation using patch priors, in: *IEEE International Conference on Computational Photography*, 2013, pp. 1–8.
- [37] R. Liu, Z. Li, J. Jia, Image partial blur detection and classification, in: *IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.
- [38] B. Su, S. Lu, C.L. Tan, Blurred image region detection and classification, in: *Proc. 19th ACM Multimedia*, 2011, pp. 1397–1400.
- [39] T. Mitchell, *Machine Learning*, McGraw-Hill, New York, NY, USA, 1997.
- [40] R.B. Palm, Prediction as a candidate for learning deep hierarchical models of data, Technical University of Denmark, DTU Informatics, 2012.
- [41] R. Yan, L. Shao, Blind image blur estimation via deep learning, *IEEE Trans. Image Process.* 25 (4) (2016) 1910–1921.
- [42] K.B. Duan, S.S. Keerthi, Which is the best multiclass svm method? an empirical study, in: *International Workshop on Multiple Classifier Systems*, 2005, pp. 278–285.
- [43] J.D. Rugna, H. Konik, Automatic blur detection for metadata extraction in content-based retrieval context, *SPIE 5304* (2003) 285–294.
- [44] L. Zhang, L. Zhang, X. Mou, et al., FSIM: A feature similarity index for image quality assessment, *IEEE Trans. Image Process.* 20 (8) (2011) 2378–2386.
- [45] W. Wang, J. Li, F. Huang, et al., Design and implementation of log-gabor filter in fingerprint image enhancement, *Pattern Recognit. Lett.* 29 (3) (2008) 301–308.
- [46] C. Schuler, M. Hirsch, S. Harmeling, Learning to deblur, *IEEE Trans. Pattern Anal. Mach. Intell.* 38 (7) (2016) 1439–1451.
- [47] P. Wieschollek, B. Scholkopf, H.P.A. Lensch, M. Hirsch, End-to-end learning for image burst deblurring, in: *Asian Conference on Computer Vision*, 2016, pp. 35–51.
- [48] S. Nah, T.H. Kim, K.M. Lee, Deep multi-scale convolutional neural network for dynamic scene deblurring, in: *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 257–265.
- [49] P. Svoboda, M. Hradí, L. Mark, P. Zemček, CNN for license plate motion deblurring, in: *IEEE International Conference on Image Processing*, 2016, pp. 3832–3836.
- [50] M. Hradí, J. Kotera, P. Zemčík, et al., Convolutional neural networks for direct text deblurring, in: *British Machine Vision Conference*, 2015, pp. 1–13.
- [51] A. Chakrabarti, A neural approach to blind motion deblurring, in: *European Conference on Computer Vision*, 2016, pp. 221–235.
- [52] W.S. Lai, J.B. Huang, Z. Hu, et al., A comparative study for single image blind deblurring, in: *Computer Vision and Pattern Recognition*, 2016, pp. 1701–1709.
- [53] W. Gao, X. Zhao, J. Zou, et al., Parametric blur estimation for blind restoration of atmospherically degraded images: class g, *Opt. Rev.* 24 (10) (2017) 1–13.
- [54] Y. Hong, G. Ren, E. Liu, et al., A blur estimation and detection method for out-of-focus images, *Multimed. Tools Appl.* 75 (18) (2016) 10807–10822.
- [55] P. Bao, L. Zhang, X. Wu, Canny edge detection enhancement by scale multiplication, *IEEE Trans. Pattern Anal. Mach. Intell.* 27 (9) (2005) 1485–1490.
- [56] L. Huang, Y.S. Xia, Blind super-resolution image reconstruction based on novel blur type identification, in: *2017 10th International Congress on CISPA-BMEI*, 2017.
- [57] M.T. McCann, T. Michael, J. Jin, et al., Convolutional neural networks for inverse problems in imaging a review, *IEEE Signal Process. Mag.* 34 (6) (2017) 85–95.
- [58] D. Eigen, D. Krishnan, R. Fergus, Restoring an image taken through a window covered with dirt or rain, in: *IEEE International Conference on Computer Vision*, 2014, pp. 633–640.
- [59] D. Kingma, J. Ba, Adam: a method for stochastic optimization, in: *International Conference for Learning Representations*, 2015.
- [60] J.P. Oliveira, M.A.T. Figueiredo, J.M. Bioucas-Dias, Parametric blur estimation for blind restoration of natural images: linear motion and out-of-focus, *IEEE Trans. Image Process.* 23 (1) (2014) 466–477.
- [61] T. Kobayashi, BFO meets HOG: Feature extraction based on histograms of oriented p.d.f. gradients for image classification, *IEEE Conf. Comput. Vision Pattern Recognit.* 2191 (9) (2013) 747–754.
- [62] S. Roth, M.J. Black, Fields of experts: A framework for learning image priors, in: *IEEE Conference on Computer Vision and Pattern Recognition*, 2005, pp. 860–867.
- [63] D. Martin, D. Fowlkes, J. Malik, A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistic, in: *8th IEEE International Conference on Computer Vision*, 2001, pp. 416–423.
- [64] K. Gu, G. Zhai, X. Yang, et al., Using free energy principle for blind image quality assessment, *IEEE Trans. Multimed.* 17 (1) (2015) 50–63.
- [65] M. Everingham, L.V. Gool, K.I. Williams, et al., The pascal visual object classes challenge, *Int. J. Comput. Vision* 88 (2) (2010) 303–338.
- [66] D. Krishnan, T. Tay, R. Fergus, Blind deconvolution using a normalized sparsity measure, in: *IEEE Computer Vision and Pattern Recognition*, 2011, pp. 233–240.
- [67] J. Kotera, F. Šroubek, P. Milanfar, Blind deconvolution using alternating maximum a posteriori estimation with heavy-tailed priors, *Comput. Anal. Images Patterns* 2 (8048) (2013) 59–66.
- [68] T. Michaeli, M. Irani, Blind deblurring using internal patch recurrence, in: *European Conference on Computer Vision*, 2014, pp. 783–798.

- [69] X. Zhou, J. Mateos, F. Zhou, et al., Variational dirichlet blur kernel estimation, *IEEE Trans. Image Process. Publ. IEEE Signal Process. Soc.* 24 (12) (2015) 5127–5139.
- [70] W. Dong, L. Zhang, G. Shi, et al., Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization, *IEEE Trans. Image Process.* 20 (7) (2011) 1838–1857.
- [71] S. Duan, K. Keerthi, Which is the best multiclass svm method? an empirical study, in: *International Workshop on Multiple Classifier Systems*, 2005, pp. 278–285.
- [72] W. Xue, L. Zhang, X. Mou, et al., Gradient magnitude similarity deviation: A highly efficient perceptual image quality index, *IEEE Trans. Image Process.* 23 (2) (2014) 684–695.
- [73] M. Cheng, N.J. Mitra, X. Huang, et al., Torr and shi-min hu global contrast based salient region detection, *IEEE Trans. Pattern Anal. Mach. Intell.* 37 (3) (2015) 569–582.
- [74] R. Dash, B. B. Majhi, Motion blur parameters estimation for image restoration, *Optik -Int. J. Light Electron Opt.* 125 (5) (2014) 1634–1640.
- [75] Z. Wang, Z. Yao, Q. Wang, Improved scheme of estimating motion blur parameters for image restoration, *Digital Signal Process.* 65 (2017) 11–18.



Liqing Huang received the B.E. and M.E. degrees in computer science from Fuzhou University, China, in 2014 and 2017, respectively. She is currently pursuing the Ph.D. degree with the School of Computer Science, Fuzhou University. Her research interests are on image and video processing.



Youshen Xia received the Ph.D degree in Automation and Computer-Aided Engineering from Chinese University of Hong Kong, China in 2000. He has held various faculty/research/visiting positions at Nanjing Post University, China, Chinese University of Hong Kong, City University of Hong Kong, Calgary University, Canada, University of Waterloo, Canada, and University of Western Australia, Australia. He is currently a Professor of the College of Mathematics and Computer Science, Fuzhou University, Fuzhou, China. His present research interests include design and analysis of neural dynamical optimization approaches, and system blind identification with applications to signal and image processing.