Single-Image Blind Deblurring for Non-uniform Camera-Shake Blur

Yuquan Xu, Lu Wang, Xiyuan Hu, and Silong Peng

Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, P.R. China

Abstract. In this paper we address the problem of estimating latent sharp image and unknown blur kernel from a single motion-blurred image. The blur results from camera shake and is spatially variant. Meanwhile, the blur kernel of motion has three degrees of freedom, i.e., translations and in-plane rotation. In order to solve this problem, we first analyzed the homography blur model for the non-uniform camera-shake blur. We simplified the model to 3-dimensional camera motion which can be accelerated by exploiting the fast Fourier transform to process subsequent image deconvolution. We then proposed an effective method to handle the blind image-deblurring problem by the image decomposition, which does not need to segment the image into local subregions under the assumption of spatially invariant blur. Experimental results on both synthetic and real blurred images show that the presented approach can successfully remove various kinds of blur.

1 Introduction

The motion blur resulted from camera shake is very common in our daily life. To solve this problem, image deblurring has recently received a lot of attention in the computer graphics and vision communities. There are many methods that try to solve this problem and some of them lead to outstanding results. However, most of these deblurring methods aim to remove the uniform motion blur of the image, which is modeled as the convolution of the sharp image and a spatially invariant blur kernel or point spread function (PSF). In the spatially invariant scenario, the blur kernel stay the same for each position of the blurred image, but this model does not often hold in practice [1] [2].

Contrarily, we deal with the observed image degraded by spatially variant blur kernels, the motion of which includes not only traditional translations along x axis and y axis but also in-plane rotations on z axis. We firstly analyze the homography blur model [3] [4] [5] [6] [7], which models the spatially-varying camera-shake blur without computing the exact blur kernel for each pixel of the blurred image. We assume the scene is in a plane that parallel to the image plane. We then simplify this blur model to delineate 3-dimensional (3D) motions of the camera which include in-plane rotations as well as translations. The simplified model can give an overall 3D kernel for the whole image. We finally recover the latent image globally rather than segment the blurred image into local subregions that are regarded as spatially invariant blurred.

K.M. Lee et al. (Eds.): ACCV 2012, Part III, LNCS 7726, pp. 336–348, 2013.

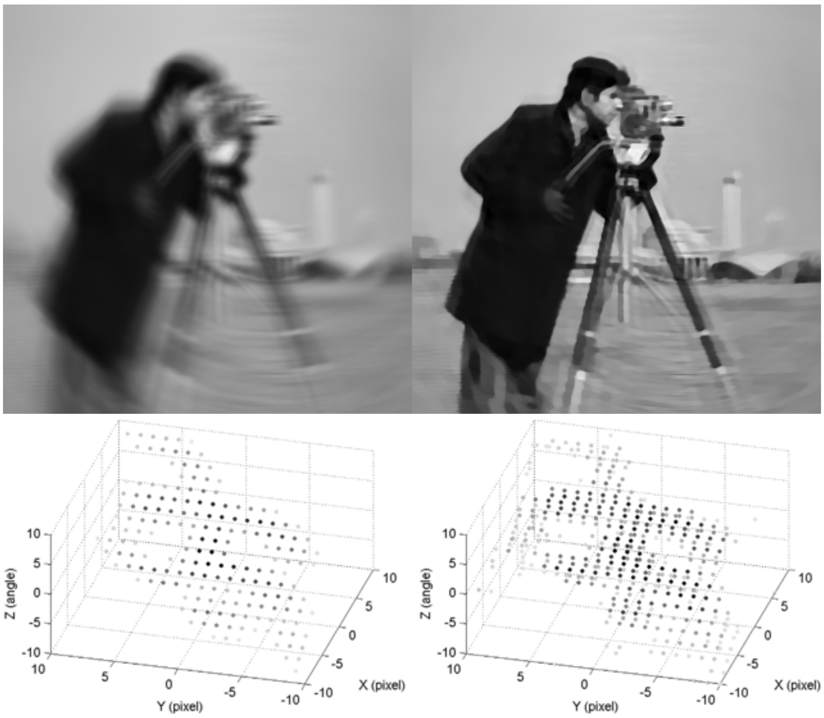

Fig. 1. Image deblurring for non-uniform motion blur. The left subfigures are the non-uniform blurred image and the ground truth 3D kernel; the blurred image is generated by convoluting the sharp Cameraman image with the kernel. The right subfigures are the deblured image and the estimated kernel using our algorithm.

There are a lot of projection transforms when using the homography blur model to treat the spatially variant image. As a result, the deblurring algorithm gets very slow and is hard to handle the blurred image with large rotation angles. Accordingly, we find that when considering the 3D motion, the homography matrix can be decomposed into two parts, the rotation and the translation, so that the numerous projection transforms in the spatially-variant deblurring problem can be simplified into some rotations of the image and a few of fast Fourier transforms (FFTs). With the help of decomposition, the image with large rotation angles can be recovered effectively. In our experiment, the treatable rotation angle of the motion reaches to more than 10 degrees. To estimate the 3D blur kernel, we utilize the trick of shock filter [8] to obtain an inference image that predicts edges of the latent sharp image, and we propose an effective method to blindly deblur the non-uniform camera-shake-blurred image. Fig. 1 is an illustration of a synthetic blurred image processed by our method.

2 Related Work

Image deblurring is a research-intensive topic of image processing. In the case of spatially invariant blur, the blur process is modeled as a convolution of the sharp image with a blur kernel or PSF. A lot of work have be done in both the non-blind convolution [9] [10] [11] [12] and blind convolution [13] [14] [15] [16]

[17] [1] [18] [19] [20] [21]. In the case of spatially variant blur, some techniques segmented the blurred image into multiple overlapped or non-overlapped subregions, each of which was of constant PSF and was deblurred individually [15] [22] [7] [23] [24]. In these schemes, the size of segmented subregions should be small enough to meet with the assumption of spatially-invariant-blurred local subregions. Meanwhile, the size of the regions should be sufficiently large to contain or even much larger than the size of blur kernels. In that way, for the blur image of large rotational angle, e.g., 10 degrees, the size of corresponding blur kernels for boundary regions of the image will be extremely huge, which will not ensure the prerequisite for spatially-invariant subregions. Tai et al. [25] [26] and Ben-Ezra and Nayar [27] took advantage of a hybrid camera to measure the non-uniform blur. Shan et al. [28] presented a method to handle the rotational blur, the range of which is required to mark by the user. Besides, there are some methods that adopted the homography blur model to deblur the image. Joshi et al. [5] explored inertial measurement sensors to catch the camera motion path during the exposure time for debluring the image. Whyte et al. [3] introduced a framework by which they recovered the 3D rotational camera motions (roll, pitch, and yaw). Tai et al. [4] proposed a projective motion-blur model to represent the non-uniform blur caused by camera shake, and used the RL algorithm with a different regularization for non-blind image deblurring. Gupta et al. [6] raised the framework of 3D camera motion (rotation on z axis and translations along x axis and y axis). They segmented the image, applied the image deblurring approach of Shan et al. [16] to each patch for an initialization, and restored the latent image. Hirsch et al. [7] combined structural constraints of the homography blur model and the efficiency of the efficient filter flow framework to obtain a fast single-image blind-deconvolution algorithm. In spite of this, their deblurring method still needs to segment the blurry image into local subregions.

3 Non-uniform Camera-Shake Blur Model

We first analyze the homography blur model and then introduce our non-uniform camera-shake blur model, which will be adopted in Section 4 to obtain our blind deblurring algorithm.

3.1 Homography Blur Model

When capturing an image, pixel intensities of the image are determined by the amount of light received by the imaging sensor over the exposure time. For a scene of static distance, the relative motion between the camera and the scene causes a planar projective transform in the image plane. Consequently, the motion-blurred image is modeled as a weighted summation of multiple images that are projectively transformed. We assume one of these poses is the original pose and the image collected in this pose is our desired latent image. In this manner, the model is formulated as

$$B = \sum_{i=1}^{n} \omega_i I_i + N \tag{1}$$

$$I_i(p_i, q_i) = I(p_0, q_0) \tag{2}$$

$$k \cdot (p_i, q_i, 1)^T = H_i(p_0, q_0, 1)^T \tag{3}$$

in which B is the observed blurred image, I denotes the latent image to be estimated, and N indicates the additive noise. Meanwhile, I_i stands for the sharp image collected in the i^{th} pose, n refers to the total number of target poses of the camera, ω_i indicates the weight and signifies the time that the camera spent in i^{th} pose, and the more the time that the camera stays in the i^{th} pose, the larger the value of ω_i; H_i is the homograph warp from I to I_i. Besides, (p_0, q_0) represent rectangular coordinates of the desired image collected in the original pose, (p_i, q_i) are coordinates of I_i, $(p_i, q_i, 1)(i = 0, 1, \cdots, n)$ stand for their corresponding homogeneous coordinates, k is the normalized factor, and ^{T} means the operation of transposition.

3.2 Our Blur Model

Formally, the homography in Eq. (3) is expressed as

$$H_i = K \left[R_i + \frac{t_i(0,0,1)}{d} \right] K^{-1} \tag{4}$$

where R_i and t_i are the rotation matrix and the translation vector, respectively, for the i^{th} pose, K is the intrinsic parameter matrix of the camera, and d is the depth of the scene. We assume d to be constant for the original pose of the camera; hence the scene is in a plane that parallel to the image plane in the original pose.

Although the camera motion still has six degrees of freedom (DOFs), we exploit only three of them in our blur model, viz., the in-plane rotation on z axis and translations along x axis and y axis. As mentioned in [3] [6], these three DOFs are already sufficient for accurately approximating the full 6D space of the camera motion. Therefore, the homography is reduced to

$$H_i = K \begin{bmatrix} \cos(\theta_z) & -\sin(\theta_z) & \frac{\Delta x}{d} \\ \sin(\theta_z) & \cos(\theta_z) & \frac{\Delta y}{d} \\ 0 & 0 & 1 \end{bmatrix} K^{-1} \tag{5}$$

where θ_z is the angle of in-plane rotation, Δx and Δy are the translational lengths along x axis and y axis, respectively; K is the intrinsic parameter matrix, i.e.,

$$K = \begin{bmatrix} f \cdot m_x & \gamma & u_0 \\ 0 & f \cdot m_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6}$$

in which f is the focal length of the camera, m_x and m_y are the scale factors that assumed to be equal as $m_x = m_y = m$, γ represents the skew coefficient between x axis and y axis and is assigned to be 0; (u_0, v_0) denote the coordinates of the principal point which are set to be the center of the image and assigned $(0, 0)$. In view of this, the final form of the homography in our model is

$$H_i = \begin{bmatrix} \cos(\theta_z) & -\sin(\theta_z) & \frac{f \cdot m \cdot \Delta x}{d} \\ \sin(\theta_z) & \cos(\theta_z) & \frac{f \cdot m \cdot \Delta y}{d} \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta_z) & -\sin(\theta_z) & dx \\ \sin(\theta_z) & \cos(\theta_z) & dy \\ 0 & 0 & 1 \end{bmatrix} \tag{7}$$

From this form, we learn that if the depth of the original scene is constant, then the depth of the images that collected in other poses are of the identical constant. For this reason, the combined parameters of $fm\Delta x/d$ can be reduced to be a single scale parameter of dx, viz., the translational pixels along x axis in the image plane; the same rule applies to dy. It means that the focal length of the image and the depth of the scene are no longer required to know. It follows that there are only three unknowns of the homography left to be determined, which are rotational angle θ_z and translational pixels of dx and dy, respectively.

In [6], Gupta et al. employed the matrix-vector form of $b = \sum_i \omega_i F_i l$, where F_i is the warp matrix that satisfies $F_i l = l_i$ and $F = \sum_i \omega_i F_i$. The matrix F_i is called the motion response basis, and was pre-computed and stored. The size of F_i is $S \times S$ and is very large, since S is the total number of pixels in the image. Given this circumstance, their algorithm would have trouble with memory when restoring seriously blurred images, the camera-motion path of which could be sampled in many discrete poses. Fortunately, in our model, it is not necessary to know the exact form of F_i or F so that our algorithm can deal with seriously blurred images without much memory. Together with the decomposition of homography matrix, the recovered image can be effectively computed.

3.3 Decomposition of the Homography Matrix

For blind image deblurring, the approximate size of the kernel is a required input from the user, due to the unknown of the exact path of camera motions during the exposure time. In our model, the kernel is in the 3D space, and the number of unknown variables of the kernel is much more than traditional 2D kernel. For example, in our synthetic experiment, the kernel size is assigned to be 21×21×21, which adds up to 9261 unknowns in the kernel. Observed from Eq. (7), we find the possibility of decomposition of the homography matrix, namely,

$$\begin{bmatrix} \cos(\theta_z) & -\sin(\theta_z) & dx \\ \sin(\theta_z) & \cos(\theta_z) & dy \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta_z) & -\sin(\theta_z) & 0 \\ \sin(\theta_z) & \cos(\theta_z) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{8}$$

The decomposition of the homography matrix means that the projective transform can be divided into two steps of the rotation and the translation.

Subsequently, the projectively transformed image I_i, whose corresponding rotational angle is θ_i and translational pixels are dx_i and dy_i, can be regarded as obtained by firstly rotating the latent image I for θ_i and then shifting the rotated image by dx_i, dy_i. That is to say,

$$b = \sum_i \omega_i F_{i,xy} F_{i,z} l \tag{9}$$

in which $F_{i,xy}$ is the warp matrix making image shifts of dx_i and dy_i, $F_{i,z}$ is the warp matrix making image rotation of θ_i. Toward the warped images that share the same rotational angle, the only distinctions among them are their individual shifts to different pixel locations. Therefore, the summation of these images can be modeled as a convolutional process and the computation can be accelerated using the FFT. In other words,

$$B = \sum_z I_{\theta_z} * \omega_z \tag{10}$$

where I_{θ_z} is the warped image by rotating θ_i, and ω_z is a 2D matrix as well as a plane in our 3D blur kernel whose corresponding rotational angle is θ_i. The blurred image thus can be calculated by several image rotations and several image convolutions. For instance, if the kernel size is 21×21×21, then only 21 rotations and 21 convolutions are needed to calculate, rather than 9621 projective transforms of the latent image. Consequently, the computational efficiency of the spatially variant deblurring problem can be greatly improved.

4 Blind Image Deblurring

The blur kernel of our model is the weights of different poses along the motion path of the camera. Our blur kernel can be expressed as a 3D matrix because of the choice of three DOFs of the camera motion. Fig. 1 has illustrated the form of the ground truth kernel, where the parameters along x axis and y axis indicate translation pixels and the parameter along z axis represents rotation angles. The non-uniform blur process is modeled as

$$B = I \odot W + N \tag{11}$$

in which W denotes the 3D kernel, and

$$I \odot W = \omega_1 I(H_1^{-1}x) + \omega_2 I(H_2^{-1}x) + \cdots + \omega_n I(H_n^{-1}x) \tag{12}$$

The procedure of our deblurring approach is divided into three steps. In each iteration, we first utilize the trick of shock filter to compute an inference image to predict edges. And then, we employ the shock-filtered image to estimate the blur kernel. Finally, we reproduce the latent image using the estimated kernel in the second step.

In [21], they use EMD to help deblur the image. Although EMD is a good method in the sinal separation but in image decomposition we use a more effective algorithm [29], which use the image decomposition in the image compression but we use it to aid the image deblur process. We decompose the blur image and use shock filter [8] to predict of edge information of the latent image. The shock filter is:

$$\frac{\partial I}{\partial t} = -sign(\Delta I)||\nabla I|| \tag{13}$$

We first decompose the image and then shock filter the cartoon part of the image. The result is used as an inference image for blur kernel estimation.

4.1 Kernel Estimation

In the step of kernel estimation, the energy function is in the form of

$$E(W) = \left(\sum_{\gamma \in \{x,y\}} ||\partial_\gamma (I_s \odot W) - \partial_\gamma B||^2 \right) + \lambda_1 ||W||_1 + \lambda_2 ||\nabla W||_1 \tag{14}$$

in which I_s is the shock-filtered cartoon part of the image, and

$$W_{optimal} = \min_W E(W) \tag{15}$$

Levin et al. [20] and Cho et al. [17] have shown that it is more efficient to estimate the blur kernel using the gradient of the image than using the image itself. Although they aim to estimate the uniform kernel, in our experiment, we find that the convergence of the 3D kernel can be speeded up by derivative space, too. The regularization of the kernel includes two parts. The first part indicates the sparsity of the kernel because the motion path of the camera, which is delineated by the blur kernel, is very sparse. The second one indicates the continuity of the kernel because the motion path forms a continuous curve in the 3D space. The gradient decent method is adopted to optimize Eq. (14), while the gradient of the cost function can be easily computed by the FFT.

4.2 Image Estimation

After the optimization of Eq. (14), we will employ the estimated kernel in current iteration to reconstruct the latent image, and the energy function is formulated as

$$E(I) = M \circ ||I \odot W - B||^2 + \lambda_3 ||\nabla I||_1 \tag{16}$$

in which M denotes a weight matrix whose values are within the range of [0, 1] and ∘ represents the element-wise multiplication operator;

$$I_{optimal} = \min_I E(I) \tag{17}$$

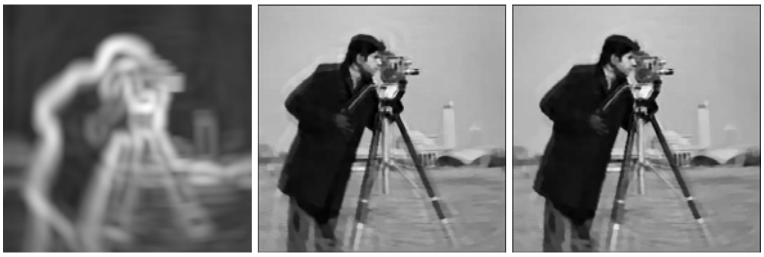

Fig. 2. The influence of the weight matrix M. The left subfigure is the matrix M calculated from the blurred image. The subfigure in the middle is the deblurred result without the matrix M in Eq. (16), and the right subfigure is the result with M in Eq. (16).

For image deblurring, the most interesting parts to reconstruct are edges and details in the latent image, and smooth parts of the blurred image are not expected to be deblurred too much in order to avoid strong artifacts such as ringing effects. In the weight matrix that added to the energy function, the larger value means that its corresponding pixel is the edge or the detail while the smaller value means the pixel is in the smooth region. To formulate this matrix, we first locate a local window centered at each pixel of the blurred image. We then compute the standard deviation of pixels for each local window, which forms a matrix of standard deviation. Finally, we normalize the standard deviation matrix into [0, 0.5], and the matrix M is right the standard deviation matrix that added 0.5. As showed in Fig. 2, the result of our algorithm contains less artifacts with the weight matrix M than the one without.We still use gradient descent method to optimize the energy function in Eq. (16). The computation of optimization can also be reduced using the FFT.

4.3 Details of the Optimization and Parameters

The boundary region of the image must be treated to avoid artifacts since we will use the FFT in image deblurring algorithm. We pad the image at the beginning of each iteration using the method in [30], and cut out the valid part of the padded image at the end of each iteration. After each iteration, the value of σ is multiplied by 0.9 to encourage more details of the image to participate in the optimization. The maximum iteration number is usually assigned to be 10, which is sufficient for attaining a high quality result as shown in Fig. 3.

5 Experiments and Results

5.1 Basic Applications

To begin with, we illustrate the performance of motion deblurring on the Cameraman image, which is convoluted with the synthetic blur kernel. The Cameraman example is the one we briefly given in the Introduction. We now compare

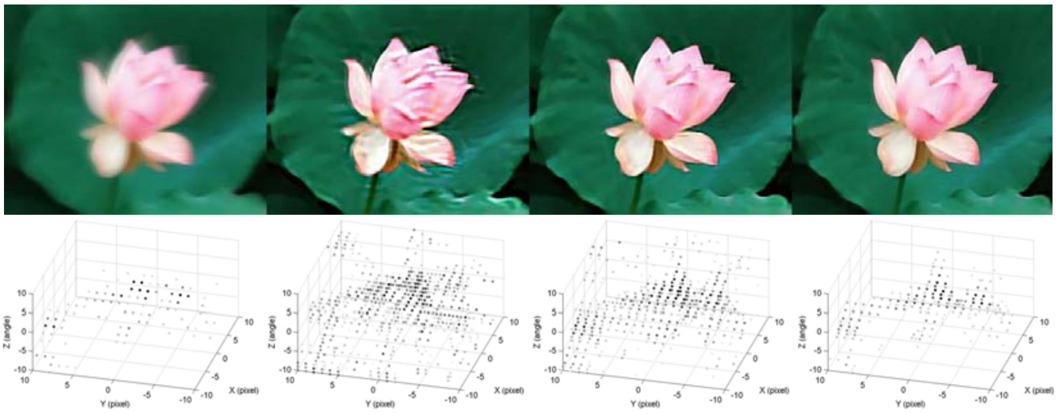

Fig. 3. Iterative results of our method. The subfigures in the first column are the blurred image and the ground-truth blur kernel, and the subfigures from the second column to the fourth column are the restored image and the estimated blur kernel in the first, the fifth and the tenth iteration, respectively.

our method against Whyte et al.'s algorithm in [3], which aims to remove the spatially varying blur, too. In their model, the blur is identified as the 3D rotational camera motion that includes rotations on x axis, y axis and z axis, respectively. This is the distinction between our blur model and their 3D rotational model. Nevertheless, if the focal length of the camera is large enough, then the blur caused by translation along and rotation on x axis and y axis have similar appearances. On such occasions, the two models can do the same work.

To make a control test, we assign the parameter of focal length in Whyte et al.'s algorithm [3] to be 1000 mm, and the experimental result is displayed in Fig. 4. The left subfigures are results of Whyte et al.'s and the right are ours. Our restoration results reveal more details and less artifacts.

We supply another synthetic example in Fig. 5, results of which are compared against those of Gupta et al.'s approach in [6]. The blur model in [6] also include 3D motions, that is, translation and in-plane rotation. In the restoration step, they segmented the image into local subregions and utilized the method in [16] to deblur each patch. They then picked up the well-deblurred patches as an initialization and took advantage of the initialization to restore the image. Seen from the ground truth kernel in the top right corner of Fig. 5, there are six DOFs of the motion in the kernel. Although our method only considers three DOFs of the kernel, it can acquire reasonable results. Compared against the result using the algorithm in Gupta et al. [6], ours have less ringing artifacts and more details.

The followings are some examples of real blurred images. We compare the performance of our method on some real-world examples presented in different blind deconvolution papers. Fig. 6 are comparisons of our results against those in [6] and [5]. The results in [6] have artifacts, e.g., in the roof of the building. While motion sensor information is required when adopting Joshi et al.'s technique, it is needless in ours.

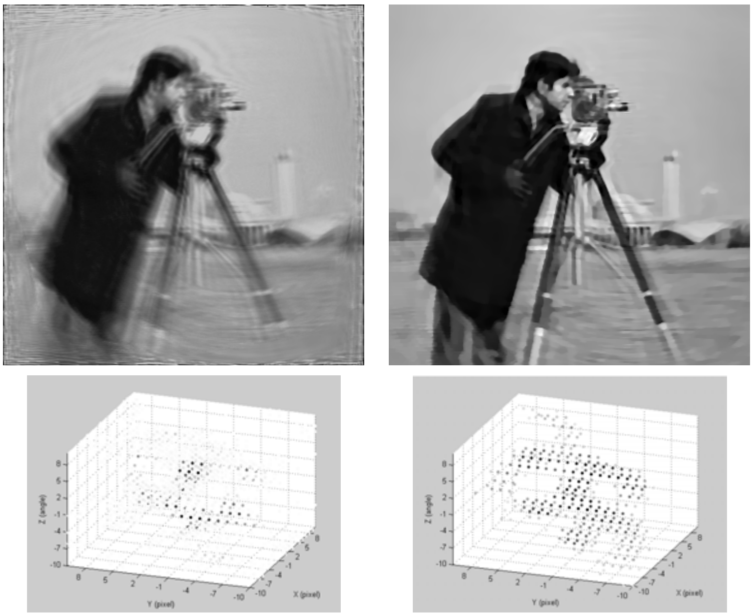

Fig. 4. Synthetic image result compared against that of Whyte et al. [3]. The left subfigures are results by the method in Whyte et al. [3], including the reconstructed image and the estimated kernel. The right subfigures are those of ours.

5.2 Extension to Zooming Blur

The blur including zooming and translations can be also viewed as a 3D blur kernel, and the homography matrix for this kind of blur switches to

$$\begin{bmatrix} 1\,0 & dx \\ 0\,1 & dy \\ 0\,0 & 1+dz \end{bmatrix} = \begin{bmatrix} 1\,0 & 0 \\ 0\,1 & 0 \\ 0\,0 & 1+dz \end{bmatrix} \begin{bmatrix} 1\,0\,dx \\ 0\,1\,dy \\ 0\,0\;\;1 \end{bmatrix} \qquad (18)$$

in which dz is the translation along z axis. In a similar manner, our method is able to remove the zooming blur, too. As shown on the right side of Eq. (18), the blur can be viewed as being produced by two individual phases, namely, the phase of convolution and the phase of zooming. It is worth noting that in the case of rotation, we perform the convolution after the rotation, but in the case of zooming, the sequence is zooming first and convolution afterwards. Because of this difference, the kernel needs to resize once the FFT is exploited for gradient calculations in the step of kernel estimation, which will bring many errors into iterations. Results of the zooming deblurring, therefore, contain more artifacts than rotational ones. Despite this, the reconstructions are still acceptable, and an example is displayed in Fig. 7.

6 Discussions and Future Work

In this work, we propose a framework of sharp image reconstruction and 3D blur-kernel identification from a single blurred image. The experimental results

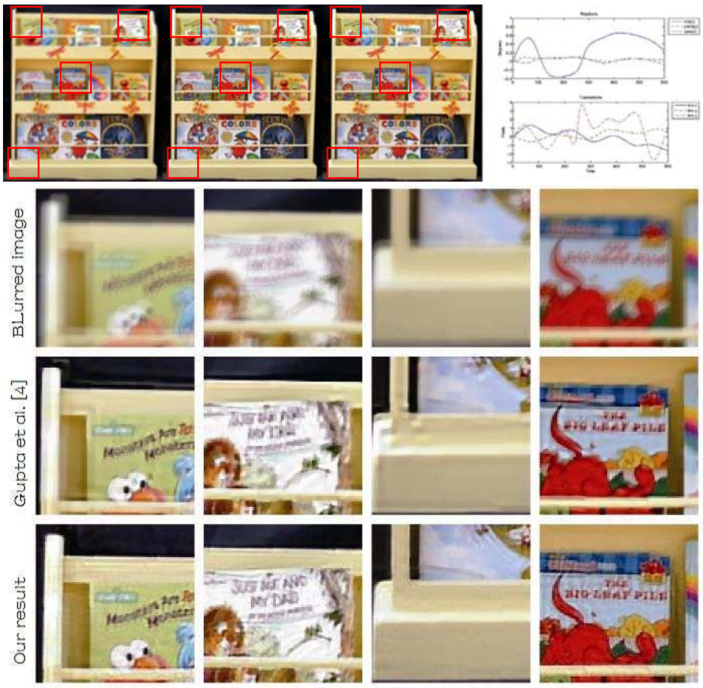

Fig. 5. Synthetic result with six DOFs of motion. The subfigures from left to right in the first row are the blurred image, followed by the result from [6], the result of our method and the ground truth kernel of the blurred image. The subfigures from the second row to the forth row are close-ups of the blurred image, deblurred image from [6] and the image deblurred by our algorithm.

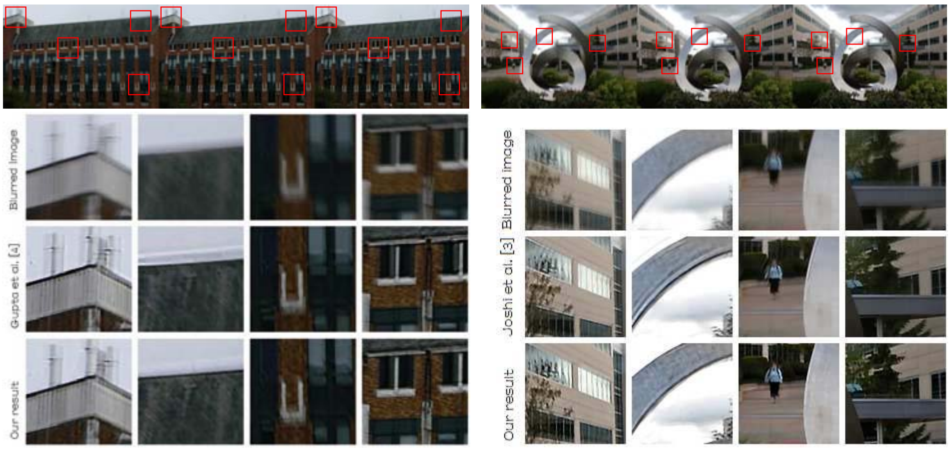

Fig. 6. Real image result compared against that of Gupta et al. [6] and Joshi et al. [5]. The subfigure from left to right in the first row are the blurred image, the result from [6] or [5] and the result by our method. The subfigures from the second row to the forth row are close-ups of the blurred image, deblurred image from [6] and [5] and the image deblurred by our algorithm.

indicate that our algorithm is competent to effectively solve the non-uniform blind-deblurring problem. Because we have exploited the FFT to reduce a lot of computations, the algorithm still will take 10 minutes for a 350×350 image the kernel size of which is 21×21×21, this image will take 5 hours using the code

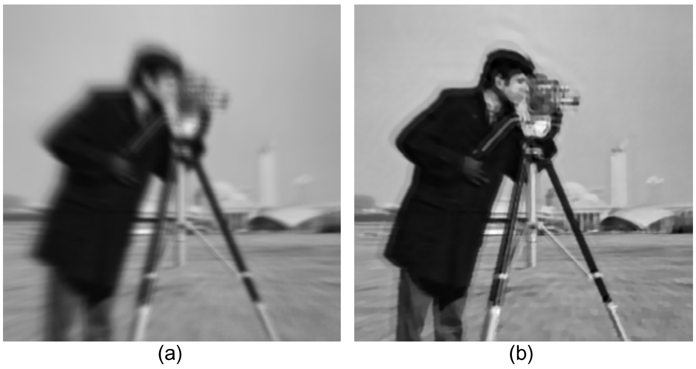

Fig. 7. Image result of zooming and translational blurs. The left subfigure is the blurred image degraded by the ground truth kernel in Fig. 1, and for the kernel on current occasion, the variable along z-axis represents the zooming scale rather than the rotation angle. The right subfigure is the reconstructed image.

in [3]. As in [7] they show that a 354×265 image with 21×21 kernel would take 724 seconds. The most computation in our algorithm is cost in the process of interpolation and the FFT.

There are several future directions to extend our work. First, other optimization algorithms will be explored to speed up the convergence. Second, the problem of how to recover the image captured from the scene of segmented depths will be addressed. We current just assume a scene of static depth, which is usually not satisfied in practice. At the same time, the reliable and flexible treatment of more kinds of blurs will be included in our blind deblurring algorithm.

Acknowledgement. This work was supported in part by the National Natural Science Foundation of China under Grant Nos. 60972126, 61101219, 61201375 and the State Key Program of National Natural Science of China under Grant No. 61032007.

References

1. Levin, A., Weiss, Y., Durand, F., Freeman, W.: Understanding and evaluating blind deconvolution algorithms. In: CVPR (2009)
2. Joshi, N., Szeliski, R., Kriegman, D.: Psf estimation using sharp edge prediction. In: CVPR (2008)
3. Whyte, O., Sivic, J., Zisserman, A., Ponce, J.: Non-uniform deblurring for shaken images. In: CVPR (2010)
4. Tai, Y., Tan, P., Brown, M.: Richardson-lucy deblurring for scenes under a projective motion path. IEEE Trans. on PAMI 33, 1603–1618 (2011)
5. Joshi, N., Kang, S., Zitnick, C., Szeliski, R.: Image deblurring using inertial measurement sensors. ACM Trans. Graph. 29, 1–9 (2010)
6. Gupta, A., Joshi, N., Lawrence Zitnick, C., Cohen, M., Curless, B.: Single Image Deblurring Using Motion Density Functions. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part I. LNCS, vol. 6311, pp. 171–184. Springer, Heidelberg (2010)

7. Hirsch, M., Schuler, C., Harmeling, S., Schölkopf, B.: Fast removal of non-uniform camera shake. In: ICCV (2011)
8. Osher, S., Rudin, L.: Feature-oriented image enhancement using shock filters. SIAM Journal on Numerical Analysis 27, 919–940 (1990)
9. Richardson, W.: Bayesian-based iterative method of image restoration. Journal of the Optical Society of America 62, 55–59 (1972)
10. Lucy, L.: An iterative technique for the rectification of observed distributions. The Astronomical Journal 79, 745 (1974)
11. Yuan, L., Sun, J., Quan, L., Shum, H.: Progressive inter-scale and intra-scale non-blind image deconvolution. ACM Trans. Graph. 27 (2008)
12. Krishnan, D., Fergus, R.: Fast image deconvolution using hyper-laplacian priors. In: NIPS, vol. 22 (2009)
13. Chan, T., Wong, C.: Total variation blind deconvolution. IEEE Trans. on Image Processing 7, 370–375 (1998)
14. Fergus, R., Singh, B., Hertzmann, A., Roweis, S., Freeman, W.: Removing camera shake from a single photograph. ACM Trans. Graph. 25, 787–794 (2006)
15. Levin, A.: Blind motion deblurring using image statistics. Advances in Neural Information Processing Systems 19, 841 (2007)
16. Shan, Q., Jia, J., Agarwala, A.: High-quality motion deblurring from a single image. ACM Trans. Graph. 27, 73:1–73:10 (2008)
17. Cho, S., Lee, S.: Fast motion deblurring. ACM Trans. Graph. 28, 145:1–145:8 (2009)
18. Xu, L., Jia, J.: Two-Phase Kernel Estimation for Robust Motion Deblurring. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part I. LNCS, vol. 6311, pp. 157–170. Springer, Heidelberg (2010)
19. Cho, T., Paris, S., Horn, B., Freeman, W.: Blur kernel estimation using the radon transform. In: CVPR (2011)
20. Levin, A., Weiss, Y., Durand, F., Freeman, W.: Efficient marginal likelihood optimization in blind deconvolution. In: CVPR (2011)
21. Wang, C., Sun, L., Cui, P., Zhang, J., Yang, S.: Analyzing image deblurring through three paradigms. IEEE Transactions on Image Processing, 1 (2012)
22. Hirsch, M., Sra, S., Schölkopf, B., Harmeling, S.: Efficient filter flow for space-variant multiframe blind deconvolution. In: CVPR (2010)
23. Harmeling, S., Hirsch, M., Schölkopf, B.: Space-variant single-image blind deconvolution for removing camera shake. In: NIPS (2010)
24. Sorel, M., Sroubek, F.: Space-variant deblurring using one blurred and one underexposed image. In: ICIP (2009)
25. Tai, Y., Du, H., Brown, M., Lin, S.: Image/video deblurring using a hybrid camera. In: CVPR (2008)
26. Tai, Y., Du, H., Brown, M., Lin, S.: Correction of spatially varying image and video motion blur using a hybrid camera. IEEE Trans. on PAMI 32, 1012–1028 (2010)
27. Ben-Ezra, M., Nayar, S.: Motion-based motion deblurring. IEEE Trans. on PAMI 26, 689–698 (2004)
28. Shan, Q., Xiong, W., Jia, J.: Rotational motion deblurring of a rigid object from a single image. In: ICCV (2007)
29. Hu, X., Xia, W., Peng, S., Hwang, W.L.: Multiple component predictive coding framework of still images. In: ICME (2011)
30. Liu, R., Jia, J.: Reducing boundary artifacts in image deconvolution. In: ICIP (2008)