Motion Deblurring for Single Photograph Based on Particle Swarm Optimization

Jing Wei
College of Information Science and Technology, Northeastern University
Shenyang, 110004, China
neujw@126.com

Zhao Hai
College of Information Science and Technology, Northeastern University
Shenyang, 110004, China
zhhai@mail.neuera.com

Song Chunhe[*]
College of Information Science and Technology, Northeastern University
Shenyang, 110004, China
tedachun@163.com

Zhu Hongbo
College of Information Science and Technology, Northeastern University
Shenyang, 110004, China
zhuhb@mail.neuera.com

Received 1 December 2010

Accepted 26 May 2013

Abstract

This paper addresses the issue of non-uniform motion deblurring for a single photograph. The main difficulty of spatially variant motion deblurring is that, the deconvolution algorithm can not directly be used to estimate blur kernel, due to the kernel of different pixels are different with each other. In this paper we firstly build up the camera pose space, and take the blurred image as the weighted summation of all possible poses of the latent image. Then the deblurring problem is converted to searching for the optimized weighted parameters in the pose space. Due to its high dimension and non-convexity we propose a framework using the particle swarm optimization algorithm to solve the problem iteratively. We also find that regions with high frequency texture may damage the deblurring process, which motivates a new latent image prediction method. A non-linear structure tensor with anisotropic diffusion and a shock filter are combined to smooth the image while keeping the salient edges of it. Experimental results show that our approach makes it possible to model and remove non-uniform motion blur without hardware support.

Keywords: image deblurring, pose space, particle swarm optimization, latent image prediction

1. Introduction

Consumer-level cameras often suffer motion blur caused by hand shake. In many situations there is not sufficient light and a long exposure is required, and if the camera is not held still the photos come out blurry. An example of blurring process is shown in Fig.1. The observed image is the integral of all poses of the latent image over the exposure time. Removing blur from a single photograph has been a fundamental research problem and received much attention in the past few years. With a few exceptions, most of current image deblurring methods assume a spatially invariant kernel, and the problem reduces to an image deconvolution

[*]Mailbox 377, Northeastern University, Shenyang, 110004, China.

issue, which is further divided into the blind and non-blind cases. If the blur kernel is known, it is a non-blind case and only a latent image must be recovered from the observed image, and if the blur kernel is unknown it is a blind deconvolution case, which is a pretty challenging and ill posed problem. The spatially invariant kernel often does not hold in practice as in fact there are many properties of a camera and a scene that can lead to spatially-varying blur such as depth dependent blur due to camera translation, roll motion, yaw and pitch motion.

The main difficulty for solving non-unform motion deblurring is that we can not directly use the deconvolution algorithm to estimate blur kernel, because the kernels of different pixels are different with others. In this paper we propose a new framework to deal with motion deblurring for a single photograph from a camera. The key idea of this work is that the observed blurred image is the integration over the image taken by the camera over all the poses in its path over the exposure, and the blurred image can be viewed as a weighted summation of all possible poses. So we can solve the deblurring problem by searching the optimized weighted parameters in the pose space.

The contributions of this work are four manifolds. Firstly, a new framework is proposed to deal with non-uniform motion blur for a single photograph, which mainly includes four steps: latent image prediction, motion estimation, kernel reconstruction, and image reconstruction. Based on the proposed framework, we can address the issue of non-uniform motion deblurring for a single photograph, even the camera motion with 6 degrees of freedom. Secondly we develop a model relating the camera motion, the latent image and the blurred image for a scene with constant depth in the pose space. Thirdly the particle swarm optimization (PSO for short) algorithm is introduced into our framework to effectively optimize the weighted parameters in the pose space, and then the result is refined by an iterative support detection method. Finally, we find that strong edges do not always help to deblur image, the region with high frequency texture may damage the deblurring process, which motivates us a new latent image prediction method. We combine a non-linear structure tensor with anisotropic diffusion and a shock filter to smooth the image while keeping the salient edges of it. This operation is especially important as we directly use the gray value of the image in the PSO algorithm.

The paper is organized as follows. In Section 2, we survey related work including non-blind deconvolution and blind decnvolution. Section 3 shows the overview of the proposed deblurring framework. From Section 4 to Section 6, we give the detail of the framework, that is, latent image prediction, kernel estimation and deconvolution. In Section 7, the results of the propose approach are compared with some state-of-the-art deblurring method. Finally in Section 8 is the conclusion.

2. Related Works

Removing blur from a single photograph has been a fundamental research problem and received much attention in the past few years. If the blur kernel is known, it is a non-blind case and only a latent image must be recovered from the observed image. And if the blur kernel is unknown, it is a blind deconvolution issue, which is a pretty challenging and ill posed problem. Dai et al. [3] propose a method to estimate spatially varying blur kernels based on values of the alpha map. Fergus et al. [4] recover a blur kernel by using a natural image prior on image gradients in a variational Bayes framework. Hirsch et al. [5] also propose a multi-frame patch-based deblurring approach but do not impose any global camera motion constraints on the spatially-varying blur. Jia [6] use transparency maps to get cues for object motion to recover blur kernels by performing blind-deconvolution on the alpha matte, with a prior on the alpha-matte. Joshi et al. [7] predict a sharp image that is consistent with an observed blurred image. Shan et al. Krishnan and Fergus [9] propose to solve the hyper-Laplacian priors by finding the roots of a cubic and quartic polynomial. Levin et al. [12] segments an image into several areas of different motion blur and then each area is deblurred independently. Yuan et al. [22] propose a progressive multi-scale refinement scheme based on an edge preserving bilateral Richardson-Lucy (BRL) method, and Wang et al [26] improved the RL algorithm based on a local prior. In [17], Raskar et al. flutter the opening and closing of the camera shutter during exposure to minimize the loss of high spatial frequencies. [18] incorporate spatial parameters to enforce natural image statistics using a local ringing suppression step. Shan et al. [19] propose a technique to handle rotational motion blur. Yun and Woo [23] proposed a linearized proximal altering minimization method for motion deblurring. They used a linearization of the fidelity term and the proximal function to efficiently solve the motion deblurring issue. Cai et al. [24] and Lakshman [25] used multiple images to deal with motion deblurred image.

In this paper we propose a new framework to handle non-uniform blur. The idea in this paper follows works of Ankit et al. [1] and Neel et al. [14]. In this work, we recover the camera motion from which the blur kernels can be derived in the pose space, rather than try to recover the spatially varying blur kernels directly. The main differences between our work and theirs are as follows, Neel et al. try to recover the camera motion based on the information from these sensors by hardware support. Ankit et al. used the framework of uniform blind deconvolution to deal with the blurred image by a motion density function. In our work we propose a new framework which searches the optimized parameters using PSO in the pose space directly. In addition, we also propose a new latent image prediction method, which use nonlinear structure tensor and shock filter to smooth the image while reconstruct salient edges. Note that some primary results have been published in [27].

3. The Proposed Method

Currently most of current image deblurring methods assume motion blur with a spatially invariant kernel, which is modeled as the convolution of a latent sharp image with a shift-invariant kernel plus noise. Blur process is commonly expressed as:

$$B = L \otimes K + N \tag{1}$$

where K is the blur kernel, N is the system noise which is typically considered to be white Gaussian noise.

Based on (1), one can get the latent image L by optimizing K and B iteratively even only has the blurred version of it. However, the spatially invariant motion often does not hold in practice [12,16], so we need to setup a more complex model of camera motion.

3.1. *Spatially variant model in the pose space*

We assume the camera initially lies at the world origin with its axes aligned with the world axes, and a camera motion is a sequence of camera poses where each pose can be characterized by 6 parameters - 3 rotations and 3 translations. During the exposure period of a camera, the intensity of light from a scene point (X,Y,Z) at an instantaneous time t is captured on the image plane at a location (u_t, v_t), which can be written as:

$$(u_t, v_t, 1)^T = P_t\left(X,Y,Z,1\right)^T \tag{2}$$

where P_t is the camera projection matrix, and it varies with the camera rotation and translation, which causes fixed points in the scene to project to different locations at each sample time, 1 in Eq.2 means that the focal length is fixed.

For an uncalibrated camera, this is a general 8-parameter homography, but in the case of a camera with known internal parameters, the homography H is parameterized rotation and translation matrix describing the rotation and the translation of the camera:

$$H(d) = \left[M\left(R + \frac{1}{d} TN^T \right) M^{-1} \right] \tag{3}$$

where M is the intrinsic matrix, R and T are the translation and rotation matrix of the camera, d is the scene depth, and N is the unit vector that is orthogonal to the image plane. Thus at sample time t, the pixel value of the image is:

$$I_t(u_t, v_t) = I\left(H_t(d)(u_0, v_0, 1)^T \right) \tag{4}$$

We rewrite (4) in matrix form as:

$$I_t = K_t(d) I \tag{5}$$

The observed image B is the integral over the exposure time T of all the warped versions of I, plus some observation noise N:

$$B = \int_0^T \left(K_t(d) I \right) dt + N \tag{6}$$

The integration of these projected observations creates a blurred image, and the projected trajectory of each point on the image plane is point's point-spread function (PSF). Thus, the spatially-varying blur estimation process is reduced to estimating the rotations R and translations T for times $[0 \cdots t]$, the scene depths d, and the camera intrinsic M. We can get the information of the camera intrinsic M in the image EXIF tags. In this work we assume d is constant because usually the customer-level camera has a long focus length, which is estimated by the method in [17].

In general, a single blurry image has no temporal information associated with it, so we can not get the exact motion path at each sample time from it. We rewrite (6) as:

$$B = \int_0^S \left(w_s K_s I \right) dt + N \tag{7}$$

In the discrete pose space, it can be formulated as:

$$B = \sum_{s=1}^{S} w_s K_s I + N \tag{8}$$

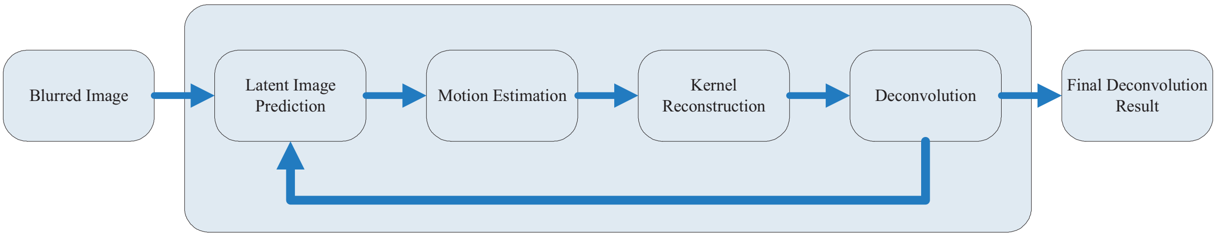

Fig.1 The framework of the proposed algorithm.

where S is the camera pose space, which consists of all the possible camera poses, w_s is corresponding parameters which indicate the time spent at the pose K_s.

All though we can form the pose space taking into account of all 3 rotation and 3 translation, but 6D pose space will make the number of the possible camera poses too huge. Oliver et al. consider that blur from camera shake is mostly due to the 3D rotation of the camera [16], while Ankit et al. show that camera motion can be modeled well by 2D translation and 1D rotation [1]. In this paper we follow Ankit et al. manner and setup a 3D pose space with 2D translation and 1D rotation.

3.2. *Process overview*

A successful approach for blind deconvolution is alternately to optimize L and K in an iterative process. In the latent image estimation and kernel estimation steps of the process, we respectively solve the equations similar to:

$$L = \arg\min_{L} \{\|B - K * L\| + \rho_L(L)\} \tag{9}$$

$$K = \arg\min_{K} \{\|B - K * L\| + \rho_K(L)\} \tag{10}$$

In (9) and (10), $\|B - K * L\|$ is the data fitting term, for which the L_2 norm is usually used, and $\rho_L(L)$ and $\rho_K(L)$ are regularization terms. In this paper we follow this manner, and Fig.1 shows the overall process of our blind deconvolution method.

To progressively refine the motion blur kernel K and the latent image L, our method iterates four steps: latent image prediction, motion estimation, kernel reconstruction and deconvolution. Compared with some previous works [18,2], we divide the kernel estimation step into two steps: motion estimation and kernel reconstruction, due to spatially-varying blur. The reason is that we can not directly use the deconvolution algorithm to estimate blur kernel because the kernels of different pixels are different with others. In our framework we can get the trajectory of the pixel at the center of the image using PSO algorithm. As we only consider the translation along the x and y direction and the rotation along the z-direction, we can get the trajectory of any pixel according to its relative position to the pixel at the center of the image.

Algorithm 1: The framework
Input:
Blurred Image B, an all-zeros kernel K, the maximal number of iterations M and the number of levels of the image pyramid N.
Build an image pyramid with level index {1,2,…,n}
For $i = 1:N$
For $M_c = 1:M$
Predict latent image (Algorithm 2);
Estimate kernel using PSO (Algorithm 3);
Refine and reconstruct kernel;
Deconvolution;
End
End
Output:
Deblurred Image, kernel for each pixel.

4. Latent Image Prediction

In some previous works [2, 21], a shock filter was used to restore salient edges in latent image. The shock filter is an effective tool for enhancing image feature, which can recover sharp edges from blurred step signals [15]. The evolution equation of a shock filter is formulated as:

$$I_{t+1} = I_t - \mathrm{sign}(\Delta I_t)\|\nabla I_t\| dt \tag{11}$$

where I_t is an image at time t, and ΔI_t and ∇I_t are the Laplacian and gradient of I_t, respectively. dt is the time step for a single evolution.

Algorithm 2: Image Prediction
Input: Blurred Image B, the current number of iteration M_c in the Algorithm 1, the maximal number of iterations M in the Algorithm1, the maximal number of iterations of the nonlinear structure tensor M_n, and the iteration number of the shock filter M_s.
$N_c = \lceil (1-M_c)/M * M_n \rceil$; % $\lceil * \rceil$ is the rounded up of * 。 For $ii = 1:N_c$ Smooth the texture of the blurred image B using equ.(12) . End For $jj = 1:M_s$ Predict edges of the smoothed image using equ.(11) . End
Output: The predicted image.

In the Algorithm 2, we use $\lceil * \rceil$ to let the minimal number of iterations of the nonlinear structure tensor be 1, in order to suppress the noise in the blurred image.

Insignificant edges make PSF estimation vulnerable to noise, as discussed in [2, 11]. But it has been found that salient edges do not always help the deblurring process. An example is shown in Fig.3. In the Fig.3, the three step signals have the same observed blurred edges, but the sparse prior always prefers the smallest intensity gradient that is consistent with the observation. Neel et al. [13] use local color statistics to provide a strong constraint during deconvolution. These constraints help to reduce over-smoothing around salient edges and high-frequency texture. Xu and Jia [21] consider that the edge information could damage kernel estimation if the scale of an object is smaller than that of the blur kernel. They select edge map for kernel estimation by measuring gradients in a sub-window. But it is difficult to determine which object's scale is smaller than the blur kernel due to the fact that we are even difficult to know what the object is. While inspired by the above two works, we consider it from another point of view, that is, if the region has high-frequency texture, information from it may mislead the deblurring process.

To overcome the issue, one can simplify the texture of the image while keep the sharp edges of it, which motivates us a new latent image prediction method. We firstly use non-linear structure tensor with anisotropic

Fig.2 Sharp edges (black) and corresponding observed blurred edges (tan). Different sharp edges may have the same observed blurred edges.

diffusion [20] to smooth the image, and then use the shock filter to reconstruct sharp edges. Vector-valued anisotropic diffusion evolves the original image under the PDE:

$$\partial_t u_i = div\left(g\left(\sum_{k=1}^{n} \nabla u_k \nabla u_k^T \right) \nabla u_i \right) \qquad (12)$$

subject to the reflecting boundary conditions:

$$\partial_v \left(g\left(\sum_{k=1}^{n} \nabla u_k \nabla u_k^T \right) \nabla u_i \right) = 0 \qquad (13)$$

where u is a vector with n components, v denotes the outer normal on the image boundary $\partial\Omega$. The diffusion time t determines the amount of simplification: when $t = 0$ the original image is recovered and larger values of t will result in more pronounced smoothing.

In our work, in the earlier iterations of the latent image prediction step, we use large iteration number of nonlinear structure tensor as at this time the latent image is far away from the original image so we can only depend on the large scale object with salient edges. Following the evolution of the latent image, we gradually reduce the iteration number of nonlinear structure tensor to allow more detail of the image to join into the motion estimation.

We give an example in Fig.3 to show the validity of our method. In Fig.3, from left to right: a blurred image; the image sharpened by the shock filter with 10 iterations; the images processed by our method, which are firstly blurred by nonlinear structure tensor with 1, 5 and 10 iterations respectively, then sharpened by the shock filter with 10 iterations. It is clear that the second image includes many small scale objects, some of which are with strong edges and can not be ignored by truncating gradient maps used in [2]. As discussed above, these small scale objects can not provide useful information but may mislead the deblurring process. While with our method, we can control the detail of the image by adjusting the number of iterations in the nonlinear structure tensor.

Fig.3: Examples of the proposed latent image prediction method. Top row: From left to right: the blurred image, the result of shock filter with 10 iterations, the rest three images: the result of the predicted image at 1st, 5th and 10th iterations. Down row: From left to right: the result of [26] and its magnified views and the result of our algorithm and its magnified views.

5. Kernel Estimation and Reconstruction

In this step we fix L and optimize w_s . The energy $E(k)$ is as follows:

$$E(k) = \left\| \nabla \left(\sum_{s=1}^{S} w_s K_s L \right) - \nabla B \right\|^2 + \lambda \|w\|^2 \qquad (14)$$

where λ is a positive parameter to balance the first item and the second item.

Optimizing w_s is difficult due to the huge number of possible poses of the camera in the pose space, and the problem is converted to searching the optimized weighted parameters in a high dimensional space. In this paper, we propose to use PSO algorithm to solve this issue. Compared to other stochastic methods such as genetic algorithm and ant colony algorithm, PSO is more suitable for the deblurring issue, because this task is to search optimized parameters in a high dimensional space with real numbers.

The PSO algorithm was first described by Kennedy and Eberhart [8]. The basic PSO (BPSO) algorithm begins by scattering a number of "particles" in the function domain space. Each particle is essentially a data structure that keeps track of its current position x and its current velocity v . Additionally, each particle remembers the "best" position it has obtained in the past, denoted p_i . The best of these values among all particles (the global best remembered position) is denoted p_g .

At each time step, a particle updates its position and velocity by the following equations:

$$v_{ij}(t+1) = wv_{ij}(t) + c_1 r_{1j}(t)(p_{ij}(t) - x_{ij}(t)) + c_2 r_{2j}(t)(p_{gj}(t) - x_{ij}(t)) \qquad (15)$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \qquad (16)$$

where $j \in \{1,2,...,D\}$, $i \in \{1,2,...,N\}$, N is the size of the population and D is the dimension of the space searched, w is the inertia weight, c_1 and c_2 are two positive constants, r_1 and r_2 are two random values into the range $[0,1]$.

Different with general optimization issue, we normalize the weight of particles at the end of the iteration, because $\|w_s\| = 1$ in the equ.(14), where

$w=\{w_s\}|_{s=1}^{D}$, D is the dimensional number of the pose space.

Algorithm 3: Kernel estimation using PSO

Input:
the maximal number of iterations of PSO M_p, the number of particles P_n, two accelerated parameters C_1 and C_2 in equ.(15), inertia parameter w, the maximal and value of position $X_{\max}$, the maximal value of velocity $V_{\max}$, kernel size $S=h*h$.

```
Initialize X_j, V_j for each particle.
For i = 1 : P_n
    Calculate Fitness using equ.(14).
    For j = 1 : S
        Update V_ij using equ.(15) .
        If V_j > V_max
            V_j = V_max ;
    Else if V_j < 0
        V_j = 0
    End
    Update X_j using equ.(16).
        If X_j > X_max
            X_j = X_max ;
    Else if X_j < 0
        X_j = 0
    End
    Update local best position;
End
Update global best position;
Normalize X_j = X_j / Σ_{j=1}^{S} X_j ;
    End
End
```

Output:
The kernel of the pixel at the center of the image.

When using PSO algorithm to optimize parameters, the first is to identify the fitness function. In our work, the fitness function is just the $E(k)$ in (14). Then we need to identify the dimension of the each particle, which is equal to the total number of possible poses in pose space. It is clearly that the number of dimension depends on the resolution of the pose space. As at each step we only want to recover the relationship between the predicted latent image and the latent image from the form iteration, we can set the maximum offset of the 1D rotation $\theta_{\max}$ and 2D translation $T_{\max}$ in each iteration to be some small values, typically, 3 degrees and 10 pixels. For translation, the resolution directly depends on the maximum offset, while for rotation we use the following method to determine its resolution. Supposing the size of image is N*M, we calculate the smallest value of rotation while drives the point on the image edge to move a pixel distance by:

$$\theta_{\min}=\arccos\left(\frac{M^2+N^2-4}{2MN}\right) \tag{17}$$

Then we set $\theta_{\max}/\theta_{\min}$ to be the basic scale of the rotation dimension. The process of optimizing weighted parameters by PSO is show as follows:

The result from PSO algorithm will have lot of small value near to zeros due to PSO use real number and hardly can make some dimensions of the particle to be zeros when the corresponding pose is independent with the blurred image, so we need refine these optimized parameters. In this paper, we use the isotropic diffusion (ISD) based Kernel Refinement proposed by Xu and Jia [21] to exclude the independent points.

With the information projected on the X-Y plane we can directly get the kernel of the center of the image, as its rotation is always zero in our pose space. We can also get the kernel of other points in the image by Eq.18. In our model, $\alpha=\beta=0$, $z'=t_z=0$, t_x and t_y are the coordinate relative to the center of the image. Then we can get the kernel of any point in the image based on the kernel of the image center and the relative coordinate of the point.

6. Deconvolution

In this step we fix K and optimize L. The energy $E(k)$ is as follows:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix}=\begin{bmatrix} \cos\beta\cos\gamma & \cos\alpha\sin\gamma+\sin\alpha\sin\beta\cos\gamma & \sin\alpha\sin\gamma-\cos\alpha\sin\beta\cos\gamma & t_x \\ -\cos\beta\sin\gamma & \cos\alpha\cos\gamma-\sin\alpha\sin\beta\sin\gamma & \sin\alpha\cos\gamma+\cos\alpha\sin\beta\sin\gamma & t_y \\ \sin\beta & -\sin\alpha\cos\beta & \cos\alpha\cos\beta & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} \tag{18}$$

$$E(L) = \|L \otimes K - B\| + \beta \|\nabla L\| \tag{18}$$

It is a non-blind deconvolution issue which contains non-linear penalties for both the data and regularization terms. This process is similar to the deconvolution in the motion deblurring framework with spatially invariant kernel [2,9,18]. In this work we mainly use the fast non-blind image deconvolution using hyper-laplacian priors proposed by Krishnan and Fergus [9].

7. Experiments and Results

For all the test in this work, we set the iteration number of shock filter 10, the maximum iteration number of nonlinear filter 20 and the minimum iteration number of it 1. For PSO algorithm, we set c1 and c2 both 1, the maximum iteration number is 50. The inertia weight w is 0.4. In the energy function, λ and β are set 1.

Fig.4 shows an example of using PSO to optimize the weighted parameters of all possible pose in pose space. Its evolution stops at 14th iteration. We compare the PSO to RANdom SAmple Consensus (RANSAC) used in [1]. Although [14] do a similar job with this paper, their performance mainly depends on the information from inertial measurement sensors. At the beginning of the evolution, the convergence speed of RANSAC is faster that that of PSO, after the 6th iteration the situation reverses. From the final convergence value it is clear that in this case PSO can find a better location than RANSAC. One can improve the algorithm by using some improved PSO algorithms, however, it is beyond the content of this paper.

Fig.5 shows some results based on some blurred images shared by [18]. These images are known to be uniform blurred, which means that the blur kernels of all pixels in these images are uniform. From Fig.5 it can be seen that results of the proposed method are similar or even better than the method of [18].

Fig.6 show our results for real-world blurred images of scenes captured using an OLYMPUS u840 camera. It shows the original blurred image, the deblurred result using spatially-variant deconvolution by the method from [1], and our deblurring result. We are deliberate to shake and rotate the camera to make the blur kernel be spatially-variation. In this situation, as shown in Fig.6, although [1] also used spatially-variant kernel, however, the method used in their work is gradient descent which is easy to fall into a local optimum value. While our approach shows a significant improvement over this approach, as PSO can search the pose space adaptively, and has more opportunities to jump out of the local optimum position.

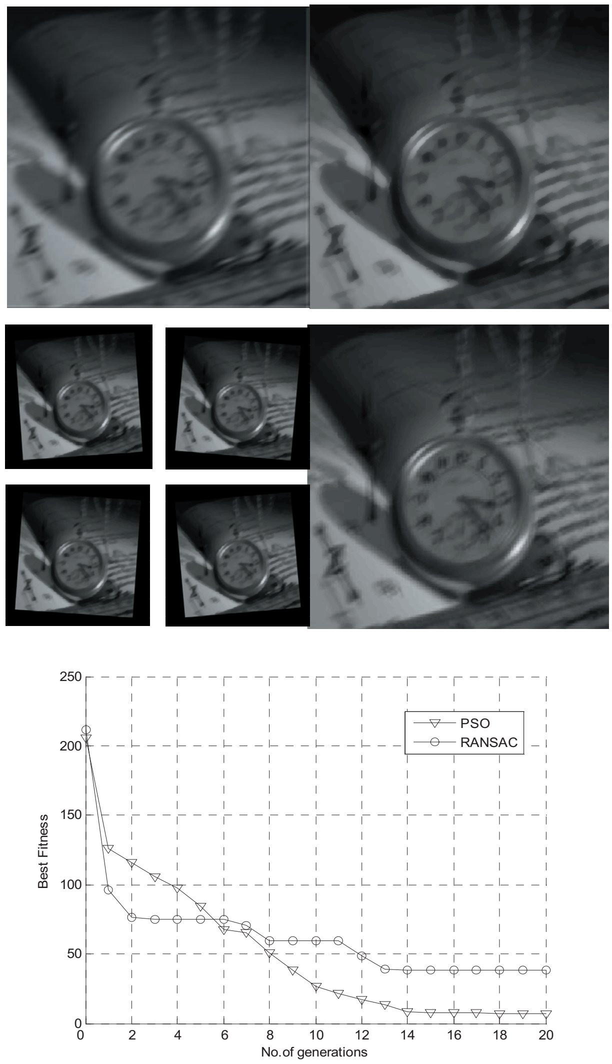

Fig.**4** One iteration in the proposed framework. First row: from left to right: the blurred image, the predicted image. Second row: from left to right: four images in pose space of the predicted image, the latent image. The third row: the convergence curves of the algorithm using PSO and RANSAC respectively. Notices that the latent image contains more sharpen edges then the blurred one.

8. Conclusion and Future Works

In this paper we propose a new framework to deal with non-uniform motion blur for a single photograph. The main idea of this paper is that we take the blurred image as the weighted summation of all possible poses of the latent image, and we develop a model relating the camera motion, the latent image and the blurred image

Fig.5 Left column: the blurred image; Middel column: deblurring results of [18]; Right column: our results.

for a scene with constant depth in the pose space. Furthermore, we find that regions with high frequency texture may damage the deblurring process and we combine non-linear structure tensor with anisotropic diffusion and a shock filter to smooth the image while keeping the salient edges of large object in the blurred image. Finally we introduce PSO algorithm into our framework to effectively optimize the weighted

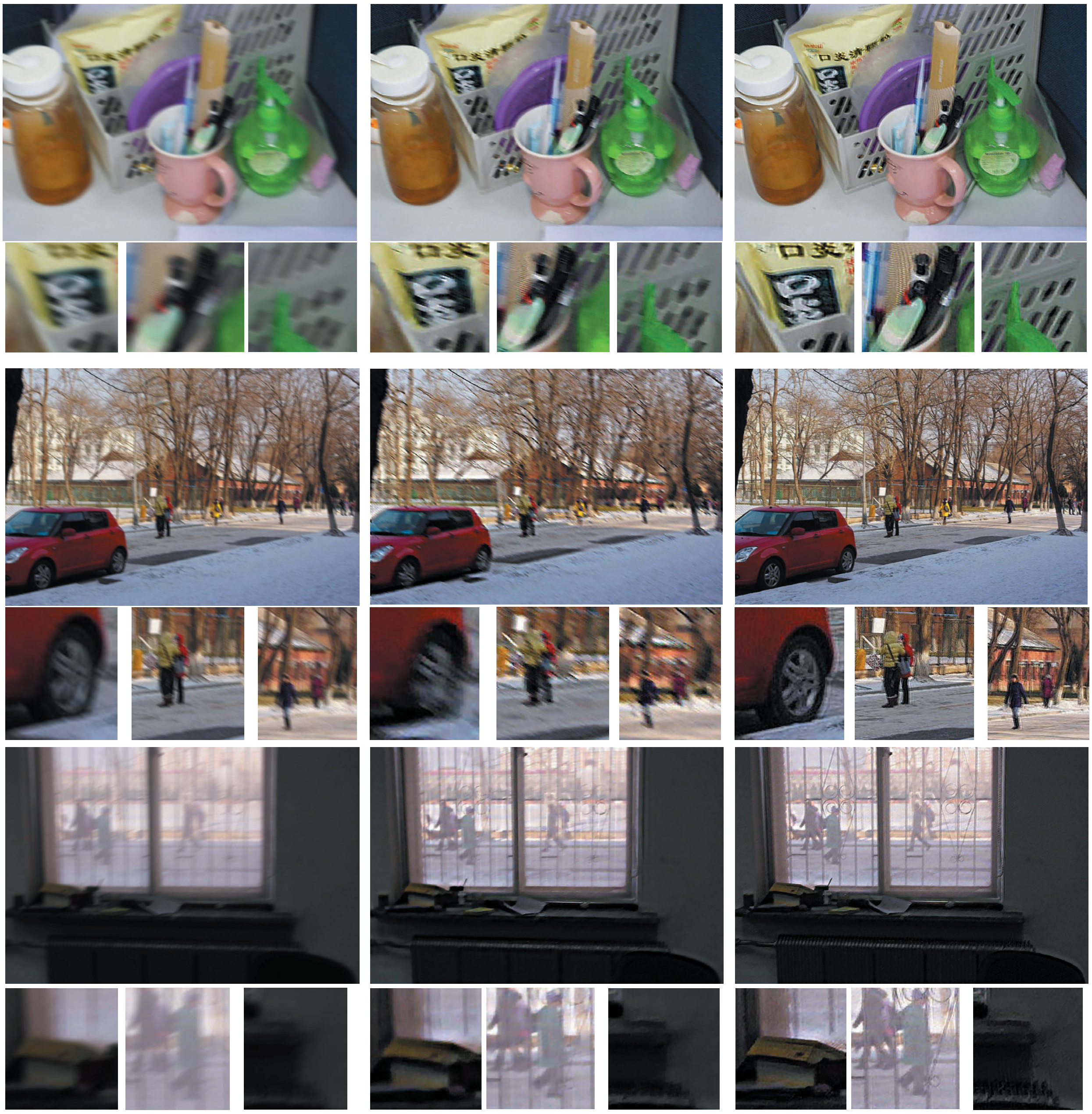

Figure 6: Left: Blurred image download from Internet. Center: The deblurring result of [1] with spatially-variant kernel. Right: The deblurring result of our work with spatially-variant kernel.

parameters in the pose space. We show that our approach makes it possible to model and remove non-uniform motion blur without any hardware support, and demonstrate its effectiveness with experiments on some challenge images.

One limitation of our method is that at the early period of the deblurring process we mainly depend on the edges of large scale objects in the predicted image. If these edges are far away from their 'true' position in the latent image, our method may fail. Another limitation is that PSO used in the framework is a random algorithm which is unstable and can not ensure convergence. In the experiments we improve the stability of the algorithm by using larger number of particles.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (50877004), and the Cultivation Fund of the Key Scientific and Technical Innovation Project Ministry of Education of China (NO708026).

Authors would like to thank anonymous reviewers for the constructive comments which greatly improve the quality of the paper.

References

1. Ankit G., Neel J., C.Lawrence Z., Michael C., Brian C.: Single Image Deblurring Using Motion Density Functions. Proc.ECCV'10 (2010), 171-184.
2. Sunghyun C., Seungyong L.: Fast Motion Deblurring. ACM TOG 28, 5 (December 2009).
3. Dai, S., Wu, Y.: Motion from blur. In: Proceedings of IEEE CVPR '08. (2008)
4. Fergus, R., Singh, B., Hertzmann, A., Roweis, S.T., Freeman, W.T.: Removing camera shake from a single photograph. ACM TOG 25, 3 (July 2006), 787-794.
5. Hirsch, M., Sra, S., Scholkopf, B., Harmeling, S.: Efficient filter flow for space variant multiframe blind deconvolution. In: Proceedings of IEEE CVPR '10. (2010)
6. Jia, J.: Single image motion deblurring using transparency. In: Proceedings of IEEE CVPR '07. (2007) 1-8.
7. Joshi, N., Szeliski, R., Kriegman, D.J.: Psf estimation using sharp edge prediction. In: Proceedings of IEEE CVPR '08. (2008)
8. J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in Proc. IEEE Int. Conf. Neural Networks, vol. 4, Perth, Australia, Dec. 1995, pp. 1942–1948.
9. Krishnan, D., Fergus, R.: Fast image deconvolution using hyper-laplacian priors.In: NIPS (2009)
10. Lucy, L. 1974. Bayesian-based iterative method of image restoration. Journal of Astronomy 79, 745-754.
11. Levin, A., WEISS, Y., DURAND, F., AND FREEMAN, W. 2009. Understanding and evaluating blind deconvolution algorithms. CVPR 2009. IEEE Conference on, IEEE Computer Society, 1964–1971.
12. Levin, A.: Blind motion deblurring using image statistics. In: Advances in Neural Information Processing Systems. (2007)
13. Neel J., C.Lawrence Z., Richard S., David J.K., Image Deblurring and Denoising using Color Priors, In: Proceedings of IEEE CVPR '08. (2008)
14. Neel J., Sing B.K., C.Lawrence Z., Richard S.: Image Deblurring using Inertial Measurement Sensors. ACM TOG 29, 30 (August 2010).
15. Osher, S., Rudin, L.: Feature-oriented image enhancement using shock filters. SIAM Journal on Numerical Analysis 27, 919–940 (1990)
16. Oliver W., Josef S., Andrew Z., Jean P.: Non-uniform Deblurring for Shaken Images. Proc.CVPR'10 (2010)
17. Raskar R., Agrawal, A., Tumblin, J.: Coded exposure photography: Motion deblurring using futtered shutter. ACM Transactions on Graphics 25, 3, 795-804 (2006).
18. Shan, Q., Jia, J., Agarwala A., High-quality Motion Deblurring from a Single Image. ACM TOG 27, 73 (August 2008), 1–10.
19. Shan, Q., Xiong, W., Jia, J.: Rotational motion deblurring of a rigid object from a single image. In: Proceedings of IEEE ICCV '07. (2007)
20. T. Brox, J. Weickert, B. Burgeth and P. Mrazek. Nonlinear Structure Tensors. Image and Vision Computing, 24, 1: 41-55 (2006)
21. Li X., Jiaya J.: Two-Phase Kernel Estimation for Robust Motion Deblurring. Proc.ECCV'10 (2010), 157-170.
22. Yuan, L., Sun, J., Quan, L., AND Shum, H.-Y. Image Deblurring with Blurred/Noisy Image Pairs. In SIGGRAPH (2007).
23. Linearized proximal alternating minimization algorithm for motion deblurring by nonlocal regularization, Pattern Recognition, Volume 44, Issue 6, 2011, Pages 1312-1326, Sangwoon Yun, Hyenkyun Woo
24. Blind motion deblurring using multiple images, Journal of Computational Physics, Volume 228, Issue 14, 2009, Pages 5057-5071, Jian-Feng Cai, Hui Ji, Chaoqiang Liu, Zuowei Shen
25. Combining deblurring and denoising for handheld HDR imaging in low light conditions, Computers & Electrical Engineering, In Press, Corrected Proof, Available online 5 November 2011, Pandya Lakshman
26. An improved Richardson–Lucy algorithm based on local prior, Optics & Laser Technology, Volume 42, Issue 5, 2010, Pages 845-849, Wang Yongpan, Feng Huajun, Xu Zhihai, Li Qi, Dai Chaoyue
27. Song Chunhe, Zhao Hai, Jing Wei, and Zhu Hongbo. PSO Based Motion Deblurring for Single Image, GECCO2011: 85-92, 2011