# PSO Based Motion Deblurring for Single Image

Chunhe Song
Information Science and Technology
Northeastern University
Shenyang, China
tedachun@163.com

Hai Zhao
Information Science and Technology
Northeastern Univerisity
Shenyang, China
zhhai@ neuera.com

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

Hongbo Zhu
College of Software
Northeastern Univerisity
Shenyang, China
zhuhb@neuera.com

## ABSTRACT

This paper addresses the issue of non-uniform motion deblurring due to hand shake for a single photograph. The main difficulty of spatially variant motion deblurring is that the deconvolution algorithm can not directly be used to estimate the blur kernel as the kernel of different pixels are different to each other. In this paper, the blurred image is considered as a weighted summation of all possible poses, and we proposed to use a PSO (particle swarm optimization) to optimize the weighed parameters of the corresponding poses after building the motion model of the camera, and an alternatively optimizing procedure is used to gradually refine the motion kernel and the latent image. The main issue of using a PSO for deblurring is that it is generally impossible to obtain the ground true of the observed blurred image, which must be used as the input of the PSO algorithm. In this paper, a non-linear structure tensor with anisotropic diffusion is used to smooth the texture while keeping the salient edges in the image. Experimental results show the validly of the algorithm.

## Categories and Subject Descriptors

I.4.3 [**Image Processing and Computer Vision**]: Enhancement-Sharpening and Deblurring;

## General Terms

Algorithms, Experimentation.

## Keywords

Computational photography, motion deblurring, PSO

## 1. INTRODUCTION

Motion blur caused by camera shake is a one of the most common problems in photography. In many situations there is not enough light, and a long exposure is required. Then if the camera is not held still the snapshots come out blurry. 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. If the blur kernel is known, it is a non-blind case and only a latent image must be recovered from the observed image.

The main difficulty for solving non-uniform motion deblurring is that we can not directly use the deconvolution algorithm to estimate blur kernel, because the kernels of different pixels are different. The key idea of this work is that the observed blurred image is the integration of 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 weighed 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 two folds. Firstly we proposed a novel deblurring framework, which can be considered as the extended version of the fast blind deconvolution proposed by Sunghyun and Seungyong [2009]. The main difference between their and our framework is that they assumed a uniform kernel and used deconvolution method to estimate it, while in our framework we assume the kernel to be non-uniform and used a PSO (particle swarm optimization) method to solve it. Secondly we develop a model relating the camera motion, the latent image and the blurred image for a scene with constant depth in pose space. Then the PSO algorithm is introduced into our framework to effectively optimize the weighted parameters in pose space. Meanwhile we find that strong edges do not always help to deblur image, since regions with high frequency texture may damage the deblurring process, which is the motivation behind a new latent image prediction method. A non-linear structure tensor with anisotropic diffusion is used to smooth the image while keeping the image's salient edges.

The paper is organized as follows. In Section 2, we survey related work including non-blind deconvolution and blind decnvolution. In Section 3 we give the overview of the proposed deblurring framework. From Section 4 to Section 7, we give the detail of the framework, that is, camera motion model, latent image prediction, kernel estimation and deconvolution. In Section 8, we show the results of our approach and finally conclude with a discussion of limitations and future work in Section 9.

## 2. RELATED WORKS

Image deblurring has received much attention in the past several decades. Deblurring is the combination of two tightly coupled sub-problems: PSF estimation and non-blind image deconvolution. If the blur kernel is known, it is a non-blind case and only a latent image must be recovered from the observed. The most famous technique may be the Richardson-lucy (RL) deconvolution [Lucy, 1974], which computes the latent image with the assumption that its pixel intensities conform to a Posisson distribution. Recently,

*GECCO' 11*, July 12–16, 2011, Dublin, Ireland.

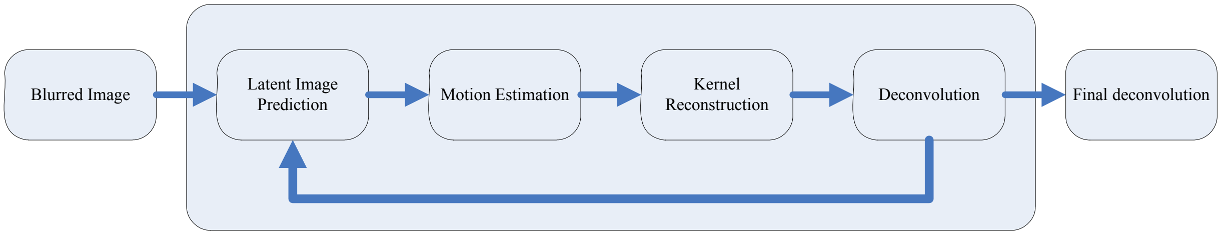

**Figure 1: Process of our deblurring framework.**

Neelamani et al [2004] propose in the signal processing community solve the deconvolution problem in the wavelet domain or the frequency domain; Donatelli et al. [2006] use a PDE-based model to recover a latent image with reduced ringing by incorporating an anti-reective boundary condition and a re-blurring step. Yuan et al. [2008] proposed a progressive multi-scale refinement scheme based on an edge preserving bilateral Richardson-Lucy (BRL) method. Krishnan and Fergus [2009] proposed to solve the hyper-Laplacian priors by finding the roots of a cubic and quartic polynomial, and using a lookup table (LUT) to achieve fast image deconvolution. Their approach can deconvolve a 1 megapixel image in less than 3 seconds.

Blind deconvolution is a significantly more challenging and ill posed problem, since the blur kernel is also unknown. Raskar et al. [2006] flutter the opening and closing of the camera shutter during exposure to minimize the loss of high spatial frequencies. Fergus et al. [2006] recover a blur kernel by using a natural image prior on image gradients in a variational Bayes framework. Yuan et al. [2007] use a pair of images, one blurry and one noisy, to facilitate capture in low light conditions. Jia [2007] 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. Shan et al. [2008] incorporate spatial parameters to enforce natural image statistics using a local ringing suppression step. Joshi et al. [2008] predict a sharp image that is consistent with an observed blurred image. They then solve for the 2D kernel that maps the blurred image to the predicted image. Levin et al. [2009] give a overview of several of these existing deblurring techniques, and show that spatial invariance is often violated, as it is only valid in limited cases of camera motion. Their experiments show that in practice in-plane camera rotation is quite common, which leads to spatially varying blur kernels.

Recently there some works on handling spatially-varying blur. Shan et al. [2007] propose a technique to handle rotational motion blur. They require user interaction for rotation cues and also rely on constraints from the alpha matte. Levin et al. [2007] segments an image into several areas of different motion blur and then each area is deblurred independently. Tai et al. [2008] developed a hybrid camera which captured a high frame rate video and a blurred image. Optical flow vectors from the video are used to guide the computation of spatially-varying blur kernels which are in turn used for deblurring. Joshi et al. [2008] run their method for non-overlapping windows in an image and use this to remove spatially varying defocus blur and chromatic aberration; however, they do not address camera motion blur, nor do they try to recover a global model of the blur. Dai et al. [2008] propose a method to estimate spatially varying blur kernels based on values of the alpha map. The method relies strongly on the pre-computation of a good alpha matte and assumes the scene to be a foreground object moving across a background. Tai et al.[2010] use a coded exposure to produce a stroboscopic motion image and estimate motion homographics for the discrete motion steps with some user interaction, which are then used for deblurring. Their method requires close user interaction and relies on non-overlapping texture information in the blurred regions. Hirsch et al. [2010] also propose a multi-frame patch-based deblurring approach but do not impose any global camera motion constraints on the spatially-varying blur.

Our approach is inspired by that of Ankit et al. [2010], Oliver et al. [2010] and Neel et al. [2010]. Similar to [2010] and [2010], we do not try to recover the spatially varying blur kernels directly, but rather recover the camera motion from which the blur kernels can be derived in the pose space. The main difference between our work and theirs is that, Neel et al. try to recover the camera motion by hardware support (3 gyroscopes and 3 accelerometers), then the main work of them is to recovery the motion path based on the information from the sensors. Ankit et al. present the blurred image by a motion density function in the pose space based on the framework of uniform blind deconvolution used in Shan et al. [2008], and then how to solve the motion density function is a hard problem. In our work we propose a new framework which searches the optimized parameters using PSO in the pose space directly. We also propose a new latent image prediction method, which use nonlinear structure tensor to smooth the image while keeping salient edges.

## 3. THE OVERALL FRAMEWORK

Blind deconvolution in this paper is alternating optimization of $L$ 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:

$$L = \arg\min_{L} \{\|B - K * L\| + \rho_L(L)\} \tag{1}$$

$$K = \arg\min_{K} \{\|B - K * L\| + \rho_K(L)\} \tag{2}$$

In (1) and (2), $\|B - K * L\|$ 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. 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.

The process of the framework in this paper is shown in Fig.1. Our framework can be considered as the extended version of the fast blind deconvolution proposed by Sunghyun and Seungyong [2009], which can effectively deblur images. The main difference between [Sunghyun and Seungyong, 2009] and our framework is that they assumed a uniform kernel and used deconvolution method to estimate it, while in our framework we assume the kernel to be non-uniform and used PSO method to solve it. Note that the reason for choosing the framework of [Sunghyun and Seungyong, 2009] is that we need the results from the predicting step as the input for the PSO based kernel estimation step in our framework as discussed in the section 5. Compared with some previous works another main difference is that we divide the kernel estimation step into two steps: motion estimation and kernel reconstruction, due to spatially-varying blur.

# 4. CAMERA MOTION MODEL

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 modeled 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.

Basing 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, so we need setup a more complex model of camera motion. We assume the camera initially lies at the world origin with its axes aligned with the world axes; 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:

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

where $P_t$ is the camera projection matrix. $P_t$ varies with the camera rotation and translation, which causes fixed points in the scene to project to different locations at each sample time.

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 matrix 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, $N$ 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 that point's point-spread function. 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 due to usually the customer-level camera has a long focus length, which is estimated by the method in [Joshi, 2010].

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 written as:

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

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. [2010] consider that blur from camera shake is mostly due to the 3D rotation of the camera, while Ankit et al. [2010] show that camera motion can be modeled well by 2D translation and 1D rotation. In this paper we follow Ankit et al. manner and setup a 3D pose space with 2D translation and 1D rotation.

In Eq.(7), except $B$, all variables are unknown. But if a latent version of $B$ is obtained, then we can estimate the motion of the camera by minimize Eq.(8):

$$E(k) = \left\|\xi\left(\sum_{s=1}^{S} w_s K_s L\right) - \xi(B)\right\|^2 + \lambda\|w\|^2 \tag{9}$$

where $\lambda$ is a positive parameter to balance the first item and the second item, $\xi(\bullet)$ is some special feature of $\bullet$, such as 1-norm or 2-norm of gradients.

## 5. LATENT IMAGE PREDICTION

Before using PSO to estimate the motion kernel, we firstly need to estimate the latent version of the blurred image. In this paper we use a latent image prediction step to do this. Note that although some previous work [Xu and Jia, 2010, Sunghyung et al, 2009] also includes the latent image prediction step, the aim of them is totally different to ours. In their work, as the motion kernel is uniform, the latent image prediction acts as an auxiliary method and not be a key point on the final results, which is why many similar works do not have this step. But in our work the motion kernel are non-uniform and the image prediction step plays a key role when solving Eq.(8).

In some previous deblurring work [Xu and Jia, 2010, Sunghyung et al, 2009], 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. The evolution equation of a shock filter is formulated as:

$$I_{t+1} = I_t - \operatorname{sign}\left(\Delta I_t\right)\left\|\nabla I_t\right\| dt \tag{10}$$

where $I_t$ is an image at time t, and $\Delta I_t$ and $\nabla I_t$ are the Laplacian and gradient of $I_t$ , respectively. $dt$ is the time step for a single evolution.

Insignificant edges make PSF estimation vulnerable to noise. But it has been found that salient edges do not always help the deblurring process. An example is shown in Fig.2, in which 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. [2008] use local color statistics to provide a strong constraint during deconvolution. Xu and Jia [2010] consider that edge information could damage kernel estimation if the scale of an object is smaller than that of the blur kernel.

In this paper we use a non-linear structure tensor with anisotropic diffusion [Brox et al., 2006] to smooth the image. The main consideration of the proposed method is to simplify the texture of the image while keeping the sharp edges of it. Vector-valued anisotropic diffusion evolves the original image under the PDE:

$$\partial_t u_i = div\left( g\left( \sum_{k=1}^{n} \nabla u_k \nabla u_k^T \right) \nabla u_i \right) \tag{11}$$

subject to the reflecting boundary conditions:

$$\partial_v \left( g\left( \sum_{k=1}^{n} \nabla u_k \nabla u_k^T \right) \nabla u_i \right) = 0 \tag{12}$$

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 a large number of iterations of the non-linear structure tensor as at this time the latent image is far away from the original image so we can only depend on large scale objects with salient edges. Following the evolution of the latent image, we gradually reduce the number of iterations of the nonlinear structure tensor to allow more detail of the image to join into the motion estimation. The detail of the latent image prediction step is shown in Algorithm 1.

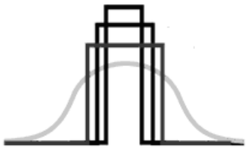

**Figure 2: Sharp edges (black) and corresponding observed blurred edges (gray). Different sharp edges may have the same observed blurred edges.**

```
Algorithm 1: Latent Image Prediction
Input:
Blurred Image $B$ , the current number of iteration $M_c$ in Algorithm 2, the maximal number of iterations $M$ in Algorithm 2, the maximal number of iterations of the nonlinear structure tensor $M_n$ .

$N_c = \lceil (1 - M_c)/M * M_n \rceil$ ; % $\lceil * \rceil$ is the rounded up of * 。
For $ii = 1 : N_c$
    Smooth the texture of the blurred image $B$ using Eq.(11) .
End
Output:
The predicted image.
```

In Algorithm 1, 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.

## 6. OPTIMIZATION USING PSO

In Eq.(9), 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 the PSO algorithm to solve this issue.

The PSO algorithm was first described by Kennedy and Eberhart [1996]. 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:

$$\begin{aligned} v_{ij}(t+1) = & \\ & wv_{ij}(t) + c_1 r_{1j}(t)(p_{ij}(t) - x_{ij}(t)) + c_2 r_{2j}(t)(p_{gj}(t) - x_{ij}(t)) \end{aligned} \tag{13}$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \tag{14}$$

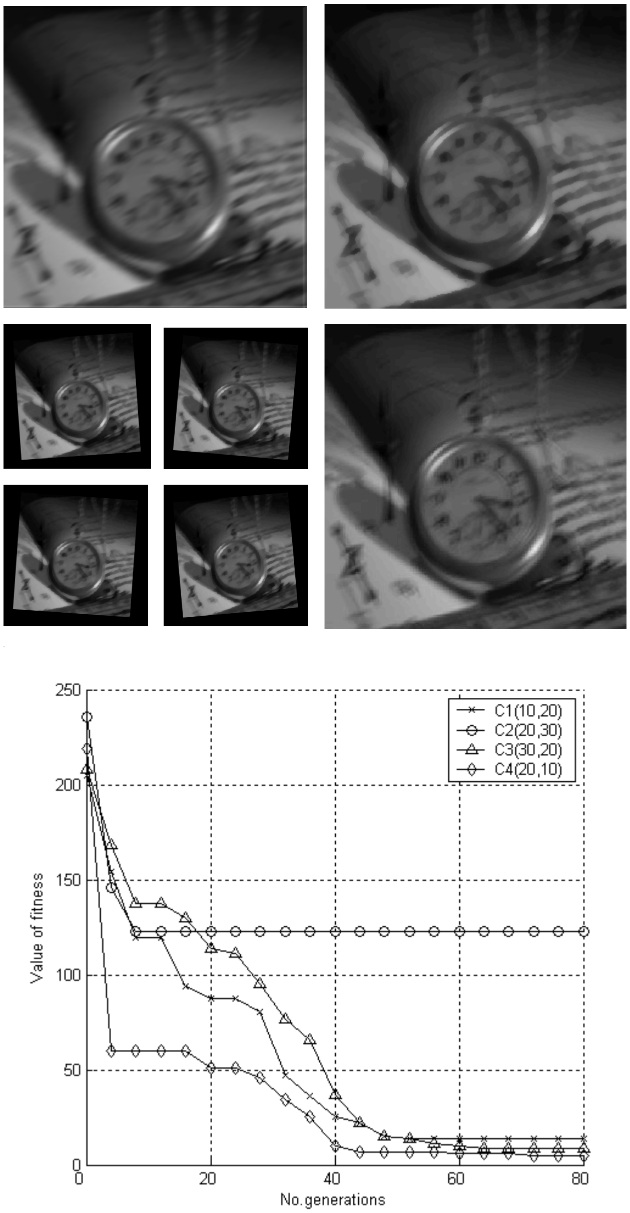

**Figure 3: One iteration in the proposed framework. First row: from left to right: the blurred image, the predicted image. Second row: from left to right: four images in pose space of the predicted image, the latent image. The third row: the convergence curves of the algorithm using PSO with different parameters respectively. Noticing that the latent image contains more sharpen edges then the blurred one.**

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 Eq.(9), where $w=\{w_s\}|_{s=1}^{D}$ , $D$ is the dimensional number of the pose space.

```
Algorithm 2: Optimization by PSO
For each particle
    Initialize xi, vi;
End
While convergence criteria is dissatisfied
    For each particle
        Evaluate the fitness of particle using (9)
        Update lbest and gbest
        For each dimension
            Update velocity and position using (13) and
            (14)
        End
    End
End
```

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 Eq.(9). 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 clear that the number of dimension depends on the resolution of pose space. As at each step we only want to recover the relationship between the predicted latent image and the latent image from the previous 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, 10 degrees and 20 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 makes the point on the image edge move a pixel distance by:

$$\theta_{\min} = \arccos\left(\frac{M^2+N^2-4}{2MN}\right) \tag{15}$$

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 shown in Algorithm 2.

Fig.3 shows an example of using PSO to optimize the weighted parameters of all possible pose in pose space. The original image is 300*300, and PSO is with 200 particles and 80 max steps. The bottom figure shows the convergence curves of PSO with different searching parameters, for example, C1(10,20) means Curve 1 with max translation 10 and max rotation 20 of the search range for PSO. The right image in the second row is the results of C4. We find that the PSO results are more sensitive to the max rotation. In this experiments, C2(20,30) can not get a satisfied result, while tests with C1(10,20), C3(30,20), C4(20,10) can yield a better results. Although we can get a better results from C2(30,20) by using more particles, but it is found to be time-consuming and not robust. And in the following tests we always use (20, 10) as the searching range for PSO. The result of the PSO algorithm will have lot of small values near to zero due to the fact that PSO uses real numbers and hardly can make some dimensions of the particle to be zeros when the corresponding pose is independent of the blurred image, so we need to refine these optimized parameters. In this paper, we use the ISD-Based Kernel Refinement proposed by Xu and Jia [2010] 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:

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

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 coordinates of the point.

# 7. DECONVOLUTION

We have already discussed the step 1, step 2 and step 3 in Section. Now we detail step 4. In the deconvolution step we fix $K$ and optimize $L$ . The energy $E(k)$ is as follows:

$$E(L) = \|L \otimes K - B\| + \beta\|\nabla L\| \qquad (16)$$

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 [Shan et al., 2008, Sunghyun and Seungyong, 2009, Xu and Jia, 2010]. In this work we mainly use the fast non-blind image deconvolution using hyper-laplacian priors proposed by Xu and Jia [2010]. In details, we denote image gradients by $q = (q_x, q_y)$ in two directions. The use of these auxiliary variables leads to a modified objective function

$$E(I, w, v) = \frac{\lambda}{2}\|I \otimes k - B\|^2 + \frac{\beta}{2}\|\nabla I - q\|_2^2 + \lambda\|q\| \qquad (17)$$

where $\beta$ is a weight that we will vary during the optimization. It is easy to see that when $\beta \rightarrow \infty$, the solution of (17) converges to that of (16). In solving for w and v given the I estimate, because w and v are not coupled with each other in the objective function, their optimization is independent. Two separate objective functions are thus yielded. We minimize (17) for a fixed $\beta$ by alternating between two steps, one where we solve for x, given values of q and vice-versa. For the x-sub problem we directly solve the following equation according to the Parseval's theorem after the Fourier transform:

$$F(I) = \frac{F(k)^* \circ F(B+v) + \frac{\beta}{\theta}\left(F(\partial_x)^* \circ F(q_x) + F(\partial_y)^* \circ F(q_y)\right)}{F(k)^* \circ F(k) + \frac{\beta}{\theta}\left(F(\partial_x)^* \circ F(\partial_x) + F(\partial_y)^* \circ F(\partial_y)\right)} \qquad (18)$$

where * is the complex conjugate and ◦ denotes component-wise multiplication.

For the q-sub problem, a auxiliary variable $v$ is introduce into the $E(L)$. In each iteration, we first compute I given the initial or estimated q by minimizing:

$$E(I, w, v) = \|I \otimes k - B - v\|^2 + \frac{\beta}{\theta}\|\nabla I - w\|_2^2 \qquad (19)$$

The *optimal solutions* for all q can be derived according to the shrinkage formula:

$$w_x = \frac{\partial_x I}{\|\nabla I\|_2}\max\left(\|\nabla I\|_2 - \theta\lambda, 0\right) \qquad (20)$$

$w_y$ can be computed similarly using the above method. Computing v can be:

$$v = \operatorname{sign}(I \otimes k - B)\max\left(\|I \otimes k - B\| - \beta, 0\right) \qquad (21)$$

where β and θ are two small positive values to enforce the similarity between the auxiliary variables and the respective terms.

# 8. EXPERIMENTS

For all the test in this work, we set the maximum iteration number of nonlinear filter 20 and the minimum iteration number of it 1. For the 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, $\lambda$ and $\beta$ are set 1. kernel size is 31*31. Fig.4 shows our results for real-world blurred images of scenes captured using an OLYMPUS u840 camera. It shows the original blurred image, the deblurred result with the method proposed by Ankit et al. [2010], and our deblurring results. The reason of choosing Ankit et al works as a comparison is that they also estimate camera motion in the pose space without hardware support. But from the experimental results it can be seen that their results are worse than us. The reasons may be as follows: firstly, although they also build up pose space and estimate camera motion, their method selects some patches with rich corners to carry out motion estimation locally without the consideration of the relationship of the pixels motion globally. Secondly, as discussed in section 5, rich texture may mislead the motion estimation process, and we used a novel latent prediction step to overcome this issue, while Ankit et al directly used the selected patch for motion estimation.

# 9. CONCLUSION

In this paper we firstly propose a novel framework to deal with motion blur for a single photograph. Secondly, 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. Thirdly, 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. Finally we introduce PSO algorithm into our framework to effectively optimize the weighted parameters in the pose space. We show that our approach makes it possible to model and remove non-uniform motion blur without any

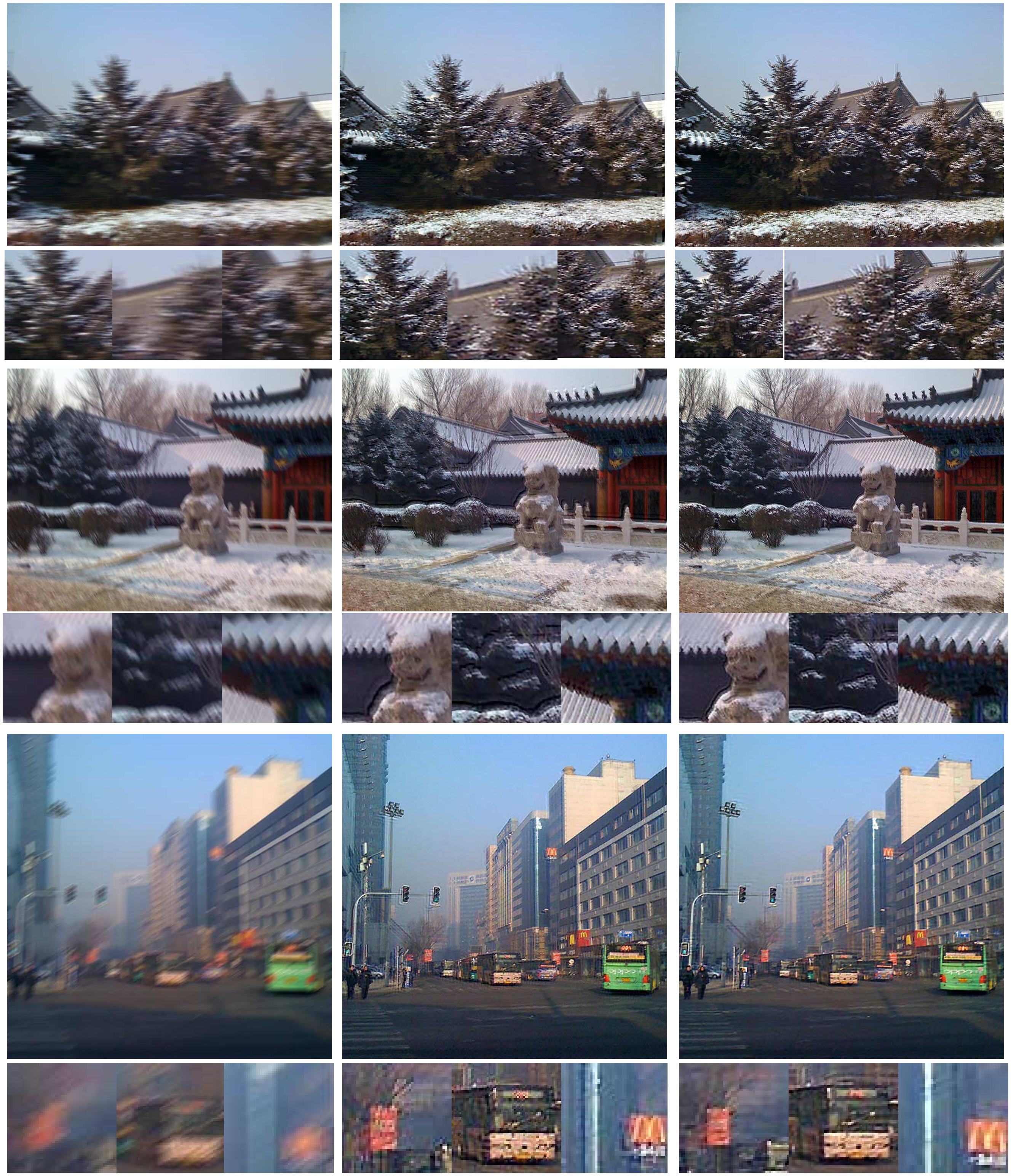

**Figure 4: Left: Blurred image. Center: The deblurring result of Ankit et al [2010]. with spatially-variant kernel. Right: The deblurring result of our work with spatially-variant kernel.**

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 the 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.

## 10. 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] Donatelli M, Estatico C., Martinelli A., Serracapizzano S.: Improved image deblurring with anti-reflective boundary conditions and re-blurring. Inverse Problems 22, 6, 2035-2053 (2006).

[4] Dai, S., Wu, Y.: Motion from blur. In: Proceedings of IEEE CVPR '08. (2008)

[5] 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.

[6] Hirsch, M., Sra, S., Scholkopf, B., Harmeling, S.: Efficient filter flow for space variant multiframe blind deconvolution. In: Proceedings of IEEE CVPR '10. (2010)

[7] Jia, J.: Single image motion deblurring using transparency. In: Proceedings of IEEE CVPR '07. (2007) 1-8.

[8] Joshi, N., Szeliski, R., Kriegman, D.J.: Psf estimation using sharp edge prediction. In: Proceedings of IEEE CVPR '08. (2008)

[9] 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.

[10] Krishnan, D., Fergus, R.: Fast image deconvolution using hyper-laplacian priors.In: NIPS (2009)

[11] Lucy, L. 1974. Bayesian-based iterative method of image restoration. Journal of Astronomy 79, 745-754.

[12] 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.

[13] Levin, A.: Blind motion deblurring using image statistics. In: Advances in Neural Information Processing Systems. (2007)

[14] Levin, A., Fergus, R., Durand, F., Freeman, W.T.: Image and depth from a conventional camera with a coded aperture. ACM Trans. Graph. 26, 70 (2007)

[15] Neelamani R., Choi H., Baraniuk R.G.: ForWaRD: Fourier-wavelet regularized deconvolution for ill conditioned systems. IEEE Transactions on Signal Processing 52, 418-433 (2004).

[16] Neel J., C.Lawrence Z., Richard S., David J.K., Image Deblurring and Denoising using Color Priors, In: Proceedings of IEEE CVPR '08. (2008)

[17] Neel J., Sing B.K., C.Lawrence Z., Richard S.: Image Deblurring using Inertial Measurement Sensors. ACM TOG 29, 30 (August 2010).

[18] Osher, S., Rudin, L.: Feature-oriented image enhancement using shock filters. SIAM Journal on Numerical Analysis 27, 919–940 (1990)

[19] Oliver W., Josef S., Andrew Z., Jean P.: Non-uniform Deblurring for Shaken Images. Proc.CVPR'10 (2010)

[20] Raskar R., Agrawal, A., Tumblin, J.: Coded exposure photography: Motion deblurring using futtered shutter. ACM Transactions on Graphics 25, 3, 795-804 (2006).

[21] Shan, Q., Jia, J., Agarwala A., High-quality Motion Deblurring from a Single Image. ACM TOG 27, 73 (August 2008), 1–10.

[22] Shan, Q., Xiong, W., Jia, J.: Rotational motion deblurring of a rigid object from a single image. In: Proceedings of IEEE ICCV '07. (2007)

[23] Tai, Y.W., Du, H., Brown, M., Lin, S.: Image/video deblurring using a hybrid camera. In: Proceedings of IEEE CVPR '08. (2008)

[24] Tai, Y.W., Kong, N., Lin, S., Shin, S.Y.: Coded exposure imaging for projective motion deblurring. In: Proceedings of IEEE CVPR '10. (2010)

[25] T. Brox, J. Weickert, B. Burgeth and P. Mrazek. Nonlinear Structure Tensors. Image and Vision Computing, 24, 1: 41-55 (2006)

[26] Li X., Jiaya J.: Two-Phase Kernel Estimation for Robust Motion Deblurring. Proc.ECCV'10 (2010), 157-170.

[27] X.Wu, B.Cheng, J.Cao, B.Cao, "particle Swarm optimization with normal cloud mutation", 7th World Congress on Intelligent Control and Automation , 2008 Pages: 2828-2832.

[28] You, Y., Kaveh, M.: Blind image restoration by anisotropic regularization. IEEE Transactions on Image Processing 8, 396–407 (1999).

[29] Yuan, L., Sun, J., Quan, L., AND Shum, H.-Y. Image Deblurring with Blurred/Noisy Image Pairs. In SIGGRAPH (2007).

[30] Y. Wang, J. Yang, W. Yin, and Y. Zhang. A new alternating minimization algorithm for total variation image reconstruction. SIAM Journal on Imaging Sciences, 1(3):248-272, 2008.