

High-Dimensional Camera Shake Removal With Given Depth Map

Tao Yue, Jinli Suo, and Qionghai Dai, *Senior Member, IEEE*

Abstract—Camera motion blur is drastically nonuniform for large depth-range scenes, and the nonuniformity caused by camera translation is depth dependent but not the case for camera rotations. To restore the blurry images of large-depth-range scenes deteriorated by arbitrary camera motion, we build an image blur model considering 6-degrees of freedom (DoF) of camera motion with a given scene depth map. To make this 6D depth-aware model tractable, we propose a novel parametrization strategy to reduce the number of variables and an effective method to estimate high-dimensional camera motion as well. The number of variables is reduced by temporal sampling motion function, which describes the 6-DoF camera motion by sampling the camera trajectory uniformly in time domain. To effectively estimate the high-dimensional camera motion parameters, we construct the probabilistic motion density function (PMDF) to describe the probability distribution of camera poses during exposure, and apply it as a unified constraint to guide the convergence of the iterative deblurring algorithm. Specifically, PMDF is computed through a back projection from 2D local blur kernels to 6D camera motion parameter space and robust voting. We conduct a series of experiments on both synthetic and real captured data, and validate that our method achieves better performance than existing uniform methods and nonuniform methods on large-depth-range scenes.

Index Terms—Blind deconvolution, motion deblurring, high-dimensional camera motion, depth dependent.

I. INTRODUCTION

CAMERA shaking is a major problem in image acquisition, especially long exposure capturing using hand-held cameras, and lots of deblurring methods have been proposed.

A. Related Works

The works on camera shake removal fall into two main streams according to the assumption on blur kernels: spatially uniform and non-uniform. The former methods formulate image blur as a 2D convolution process, and perform deconvolution on a single blurred image [1]–[11], multiple images

Manuscript received January 8, 2012; revised May 23, 2012, August 31, 2012, December 8, 2012, June 23, 2013, and November 20, 2013; accepted February 24, 2014. Date of publication April 25, 2014; date of current version May 13, 2014. This work was supported by the Project of NSFC (No. 61035002, 61171119) and Tsinghua National Laboratory for Information Science and Technology (TNList). The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Farhan A. Baqai.

The authors are with the Department of Automation, Tsinghua University, Beijing 100084, China (e-mail: yuetao.thu@gmail.com; jisuo@tsinghua.edu.cn; qionghaidai@tsinghua.edu.cn).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TIP.2014.2320368

captured at different settings such as long-exposure/short-exposure image pair [12] [13], or an image set deteriorated by different blur kernels [14]–[18]. The uniform methods assume that all the pixels are blurred with the same blur kernel, but this assumption does not hold because the image blur at a specific position is highly correlated with both camera motion and corresponding scene depth, i.e., spatially varying [19]. Therefore, *more and more researchers focus their attentions on non-uniform motion blur recently*. As discussed by Joshi *et al.* [20], the spatial variance of Point Spread Functions (PSFs) are correlated with two types of camera motion, rotation and translation, between which the spatial variance caused by rotation is depth independent, while that caused by translation heavily depends on scene depth.

For the *rotation* caused non-uniform blur, Shan *et al.* [21] propose an in-plane rotation blur model. As an improvement, a 3-DoF rotation blur model is proposed by Whyte *et al.* [22], who take all the 3 rotation directions into consideration. However, all these works only consider camera rotation during exposure and thus cannot handle camera translation, which is also a typical camera motion and causes significant image blur. Since blur caused by camera translation is depth dependent [20], Favaro *et al.* [23] and Sorel *et al.* [24] and Xu *et al.* [25] explicitly compute scene depth by stereo approach to address the image blur caused by camera translation. Taking *both translation and rotation* into consideration, Joshi *et al.* [20] use inertial sensors to measure the 6-DoFs camera motion, and perform image deblur subsequently. With a similar model, Tai *et al.* [26] estimate 6-DoF camera motion from labeled cues before applying deblur algorithm. Without external input, Gupta *et al.* [27] use in-plane rotation and x , y -translation to approximate the 6-DoF motion to reduce computational cost and use a blind deblur algorithm to restore the sharp image. Furthermore, Harmeling *et al.* [28] and Hirsch *et al.* [29] gives an efficient-filter-flow based deblurring framework which can deal with both rotation and translation fast. Lee *et al.* [30] combine the image restoration algorithm with Simultaneous Localization And Mapping (SLAM) technique to solve their 6-DoF motion blur model which assumes that the local blur kernels are straight lines. In spite that being able to handle both camera rotation and translation and give promising results, there are two major limitations to above methods: firstly, they either depend on external input (hardware or user interaction) for camera motion estimation/acquisition or simplify the motion parameters to avoid intensive computation; secondly, most of the above blur models assume planar scenes.

There are also some previous work using high-rate low-resolution/low-rate high-resolution hybrid camera systems to remove non-uniform image blur due to camera shake, such as [31]–[33]. All these methods compute optical flow from high-framerate image sequences to approximate the blur kernel instead of computing camera motion explicitly. However, the performance is prone to estimation error of optical flow because of lacking a unified camera motion as constraints.

As described before, depth acquisition is of crucial importance for camera motion deblurring. Fortunately, depth acquisition has been a significant research topic over the past decades and made great progress. Several methods like structure from motion [34], structured illumination [35] and others which we collectively called shape from X are proposed. Recently, real-time depth camera is invented and has been used in [36]–[38] successfully. Either based on time of flight [39] or coded illumination [40], currently available depth cameras perceive scene depth by active illumination and work well even in low light environment. In sum, capturing scene structure becomes easier with higher accuracy, and would provide necessary information for solving depth-aware deblurring problems.

B. Our Approach and Paper Organization

To deal with the non-uniform blur caused by an arbitrary camera motion, both scene structure and high-dimensional motion need to be considered. In this paper, we use a depth-aware projective blur model considering scene depth and describes motion with 6 DoFs explicitly. Practically, scene structure can be acquired by a depth camera or computed by structure from X approaches, while computing high-dimensional motion parameters will severely intensify the computation cost, which is a common problem in non-uniform deblurring. Actually, computing 6-DoF motion parameters using traditional methods is intractable, so most of the existing methods simplify the motion blur model to lower DoFs to approximate the 6D camera motion. To the best of our knowledge, state-of-the-art blind deblurring approaches can only handle camera motion of no higher than 3 DoFs, which limits the capacity of these algorithms for dealing with the image of large-depth-range scenes deteriorated by 6D camera motion.

To overcome this problem, Temporal Sampling Motion Function (TSMF) and Probabilistic Motion Density Function (PMDF) are proposed to reduce variable number and improve the convergence respectively.

1) TSMF: Traditional blind deblurring methods use the probability density function of motion parameter to describe the camera motion by sampling uniformly in parameter space and giving each sample a weight to describe the fraction of time the camera spent on this discretized pose. As the parameter dimension increases, the needed sample size increases drastically and induces high computational cost. In addition, because camera trajectory is just a 1D curve in the 6D parameter space, most of the samples in parameter space are trivial (zero entries), which implies that this sampling method is inefficient. Furthermore, since the sparsity constraint of

the motion has to be used to regularize the problem, this will further intensify the computation cost. To address this problem, TSMF is proposed to describe camera motion by sampling camera poses in time-domain and each sample needs at most 6 parameters to describe the camera pose at this moment. Without sampling in the triviality area of parameter space, TSMF is much more efficient and thus reduces the variable number considerably.

2) PMDF: Deblurring approaches usually optimize camera motion and latent sharp image iteratively, however, the convergence of high-dimensional motion blur model is very difficult. In this paper, PMDF is adopted to constrain the motion parameters and thus improves convergence of optimization. In computation aspect, we propose to compute PMDF by a robust voting framework from low-dimensional blur kernels, which can be estimated from local image patches. In practice, we describe the PMDF in a probabilistic manner instead of an exact optimum to raise the robustness to estimation error of low-dimensional blur kernels.

This paper firstly describes the adopted imaging model and parametrization (TSMF) in Sec. II, and then gives the two steps of our algorithm in Sec. III and IV respectively:

1) Compute PMDF

- Split image into patches and estimate their 2D local blur kernels
- Calculate the confidence of 2D local blur kernels
- Project 2D local blur kernels back to 6D parameter space and estimate PMDF by robust voting

2) PMDF guided camera shake removal

- Add PMDF to objective function as a constraint
- Iteratively optimize TSMF and sharp image

After the experiments for model analysis and validation in Sec. V, this paper concludes with some discussions.

In summary, the proposed model is advantageous over the previous methods in multiple aspects: (i) depth and 6-DoF camera motion are both explored explicitly to address arbitrary motion blur for large depth range scene; (ii) camera motion is modeled completely with 6 DoFs, and TSMF is proposed to reduce the scale of the problem effectively. (iii) PMDF is proposed to impose a unified constraints to spatially varying blur, and it can be computed effectively from low-dimensional local kernel estimation under a robust voting scheme.

Note that this paper assumes a given depth map, which may limit the application of our method to some extent. However, in Sec. V, we show that the depth map can be derived easily by depth sensor or multi-view stereo methods in many cases.

II. DEPTH-AWARE PROJECTIVE BLUR MODEL

In this section, we establish our depth-aware projective blur model which describes 6-DoF camera motion during exposure and scene depth explicitly.

In this paper, we denote the camera motion during exposure time $[t_s, t_e]$ as $\mathbf{M}(t)$. At each instant, the camera pose is represented with at most 6 DoFs— $\{T_x, T_y, T_z, \theta_x, \theta_y, \theta_z\}$ which represent translation along and rotation around x -, y -, z -axes (see Fig. 1(a)) respectively. Our method allows users to choose all the DoFs or some of them according to specific

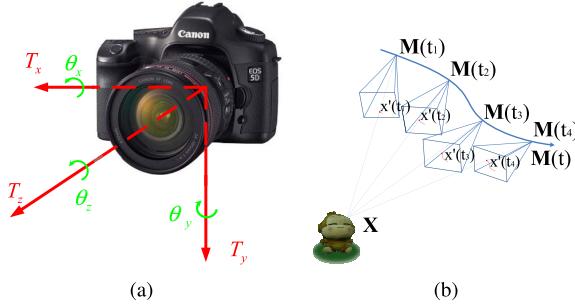


Fig. 1. Diagram of our projective blur model. (a) 6 DoFs of camera motion considered in our model. (b) The illustration of TSMF.

cases flexibly. Mathematically, $\mathbf{M}(t)$ is an element in a 6D continuous function space or its subspace:

$$\mathbf{M}(t) \in \mathbb{C}[t_s, t_e]^{Dim}, 2 \leq Dim \leq 6. \quad (1)$$

Here \mathbb{C} is the continuous function space defined over exposure time interval $[t_s, t_e]$. To facilitate representation and derivation, translation is represented by a vector $\mathbf{T} = [T_x, T_y, T_z]'$, and rotation in axis-angles is transformed to a matrix by matrix exponential $\mathbf{R} = e^{[\theta]} \times$, see [41] for details.

Because of camera shake during exposure, a single scene point projects to multiple pixels along a continuous trajectory, which depends on both the 3D coordinate of the scene point and camera motion during exposure. Fixing the focal length, the projection at time t can be expressed by

$$\mathbf{x}'(t) \sim \mathbf{K}(\mathbf{R}(t)\mathbf{X} + \mathbf{T}(t)), \quad (2)$$

where $\mathbf{x}'(t)$ is the point spread trajectory (we borrow the concept of PSF) in image plane of scene point \mathbf{X} , \mathbf{K} is the camera intrinsic matrix [41] that is assumed to be time-invariant during exposure, $\mathbf{R}(t)$ and $\mathbf{T}(t)$ respectively denote the rotation matrix and translation vector at time t .

Setting the camera pose corresponding to latent image as reference pose ($\mathbf{R} = \mathbf{I}_{3 \times 3}$, $\mathbf{T} = \mathbf{0}$), \mathbf{X} can be directly calculated from its corresponding pixel on latent image $\mathbf{x} = (u, v, 1)'$ (represented by homogenous coordinate) and the depth of that pixel $D(u, v)$ (denoted by D for short):

$$\mathbf{X} = D\mathbf{K}^{-1}\mathbf{x}. \quad (3)$$

Substituting Eq. 3 into Eq. 2, we can define a mapping between a pixel \mathbf{x} on the latent image and its corresponding pixel on imaging plane $\mathbf{x}'(t)$ at time t . The mapping depends on both the camera motion \mathbf{M} and scene depth \mathbf{D} simultaneously. Here, we denote this mapping as:

$$\begin{aligned} \mathbf{g}_{\mathbf{M}(t), \mathbf{D}} : \mathbb{R}^2 &\rightarrow \mathbb{R}^2 \\ \mathbf{x} &\mapsto \mathbf{x}'. \end{aligned} \quad (4)$$

As is known, the observed image \mathbf{I} can be represented by an integration within the exposure time range, then we have:

$$I(u', v') = \frac{1}{t_e - t_s} \int_{t_s}^{t_e} L(u(t), v(t)) dt + n(u', v'), \quad (5)$$

where the observed image \mathbf{I} is assumed to be contaminated by the i.i.d. noise $n(u', v')$, $\frac{1}{t_e - t_s}$ is a normalizing factor to ensure that observed image \mathbf{I} and latent image \mathbf{L} are at the same

intensity level. The trajectory $\{u(t), v(t)\}$ is the integration path, which can be derived by the inversion of transformation $\mathbf{g}_{\mathbf{M}(t), \mathbf{D}}$, then Eq. 5 becomes:

$$I(u', v') = \frac{1}{t_e - t_s} \int_{t_s}^{t_e} L(\mathbf{g}_{\mathbf{M}(t), \mathbf{D}}^{-1}(u', v')) dt + n(u', v'). \quad (6)$$

Eq. 6 is our depth-aware blur model describing both 6-DoF camera motion $\mathbf{M}(t)$ and scene structure \mathbf{D} explicitly.

For computing convenience, the above blur model needs to be parameterized. Traditional methods suppose each DoF has N_p samples, then N_p^6 (usually $10^6 \sim 10^8$) variables are needed to describe the camera motion. In this paper, Time-Sampling Motion Function (TSMF) (denoted by \mathbf{M}_t , $t = 1, 2, \dots, N_t$) divides exposure time into small time slices, during which the camera pose can be assumed to be constant and samples a series of camera poses to describe the camera motion, as shown in Fig. 1(b). Therefore, we can use at most 6 parameters to describe a sample point, i.e. $6N_t$ (usually $10^3 \sim 10^4$) variables are need in case of N_t samples, so the parameter number is reduced dramatically compared to traditional parameterization. In our experiments, approximately hundreds of samples are sufficient. In addition, the sampling process is performed along time axes, so the sparsity constraint is implied in the representation and does not need to be introduced into the optimizational objective function explicitly.

III. HIGH-DIMENSIONAL PMDF

Camera motion estimation in blind deblurring is a non-convex problem, and it becomes more difficult to solve when the dimension of parameter space grows. This paper propose to estimate the high-dimensional (> 2) motion by projecting low-dimensional projections (2D kernel) back to high-dimensional motion parameter space. Considering the estimation error of 2D kernel, we compute a Probabilistic Motion Density Function (PMDF) which represents the probability distribution of camera motion in high-dimensional parameter space. Then PMDF is utilized as guidance in the following optimization step. We will explain the two key steps of PMDF calculation, which are illustrated in Fig. 2. To facilitate representation, in the diagram, we use 3D motion blur model with translations on x , y axes and rotations around z axes. The 3D PMDFs are represented by 3D cubes with different density in different colors, and each cube in the 3D space corresponds to a tuple of motion parameters.

A. Patch-Based Local 2D Kernels Estimation & Confidences

In this step, we uniformly crop overlapping image patches with constant patch size from the input blurred image to estimate local blur kernels. The patch size can be determined according to the camera intrinsics and maximum motion range by considering the rotation caused non-uniformity. The overlap size is set to be slightly larger than the maximum 2D kernel size. Then, in this paper the fast deblurring algorithm [42] is applied to the patches to estimate the local blur kernels. Specifically, from coarse to fine scales, the bilateral and shock filters are used to predict the latent image, followed by the kernel estimation and deconvolution (See [42] for

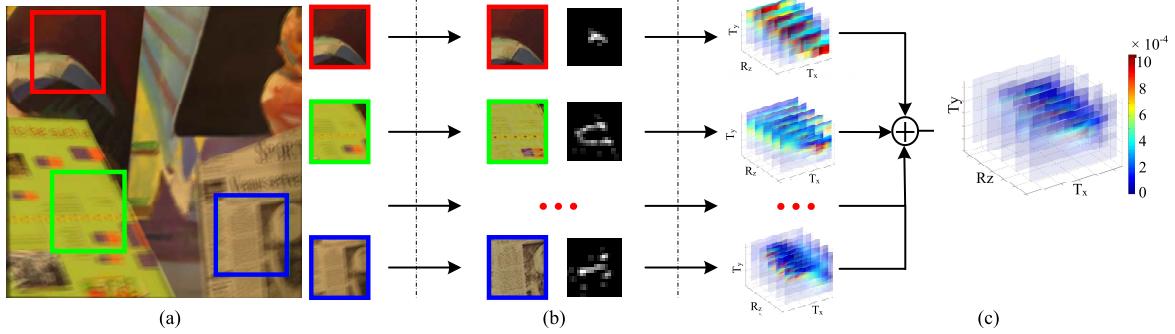


Fig. 2. The diagram of our PMDF computation. (a) Input image and several selected image patches. (b) Estimated local blur kernels and sharp image by uniform blind deblurring methods. (c) Estimating the PMDF from local kernels by back projection and robust voting. The 3D PMDFs are represented by 3D cubes in (b)(c) with different densities by different colors.

details). Note that, any uniform deblurring algorithms can be used to replace the adopted algorithm. Finally, each patch is given a confidence according to its intensity and depth by considering the translation caused non-uniformity. Afterwards, this confidence is used to form a robust estimator of high-dimensional PMDF.

1) *Size of Patches*: Generally, conventional deblurring methods work better in bigger patches. However, considering the non-uniformity of blur kernels within a single patch, we should limit the patch size to ensure that the uniform assumption can approximate the real case well. Because translation caused non-uniformity mainly depends on scene structure rather than patch size, we only discuss rotation caused non-uniformity here. According to the camera intrinsics and range of rotation parameters, we can determine the maximum patch size under a certain tolerance of non-uniformity within a patch.

First, we define the non-uniformity metric between two point spread trajectories $\mathbf{x}'(t)$ and $\mathbf{y}'(t)$ at coordinate \mathbf{x} and \mathbf{y} respectively, then the metric can be defined by:

$$\text{metric}(K(\mathbf{x}), K(\mathbf{y})) = \max_t \|d\mathbf{x}'(t) - d\mathbf{y}'(t)\|_2, \quad (7)$$

where $d\mathbf{x}'(t) = \mathbf{x}'(t) - \mathbf{x}$ and $d\mathbf{y}'(t) = \mathbf{y}'(t) - \mathbf{y}$. For a patch centered at \mathbf{x} and of radius $\Delta\mathbf{x}$, let the tolerance of non-uniformity being ϵ , we have

$$\max_{\Theta} \|\langle \mathbf{g}(\mathbf{x} + \Delta\mathbf{x})|_{\Theta} - (\mathbf{x} - \Delta\mathbf{x}) \rangle - \langle \mathbf{g}(\mathbf{x})|_{\Theta} - \Delta\mathbf{x} \rangle\|_2 < \epsilon. \quad (8)$$

Assuming Θ being limited in a bounding box with vertex set \mathcal{V}_{Θ} , the left side of Eq. 8 is maximized at one of the vertices. Performing Taylor expansion at \mathbf{x} yields

$$\max_{\Theta} \|\langle \mathbf{g}'(\mathbf{x})|_{\Theta \in \mathcal{V}_{\Theta}} \Delta\mathbf{x} - \Delta\mathbf{x} \rangle\|_2 < \epsilon, \quad (9)$$

then the maximum patch size satisfies:

$$\|\Delta\mathbf{x}\|_2 < \frac{\epsilon}{\|\langle \mathbf{g}'(\mathbf{x})|_{\Theta \in \mathcal{V}_{\Theta}} - 1 \rangle\|_2}. \quad (10)$$

Note that the patch size derived by Eq. 10 changes with position \mathbf{x} , so we uniformly sample several points in the image and choose the minimum patch size computed as the constant size for patch cropping. In Eq. 10, there is no unknowns except for non-uniformity tolerance ϵ , camera settings and motion range, which can be computed by camera intrinsics and maximum

2D kernel size. Tolerance ϵ is set to be 2 empirically, so we need only get the camera intrinsics and estimate the maximum size of 2D local blur kernels to set the patch size. In addition, since we only considered the rotation caused non-uniformity, for the scene with intensive depth variation, the patch size should be tuned for better performance.

2) *Confidence of Patches*: There are two main factors affecting the accuracy of estimated local blur kernels: richness of texture and uniformity of blur kernels within the patch. In the following, we will discuss the confidence calculation of estimated local blur kernels according to these two factors.

a) *Richness of texture*: It is reasonable that blur kernel estimated from a local image patch with abundant texture should be more creditable than that from textureless patch. Therefore, the traditional texture measurements like image entropy or Harris corner metric can be used in our scenario. Here we adopt image entropy defined as:

$$C_H(\text{patch}) = - \sum_{i,j} f_{ij} \log f_{ij}, \quad (11)$$

where f_{ij} is the frequency of pixels in the patch with intensity being i and the mean value of its neighborhood being j .

b) *Uniformity of blur kernels*: Although the rotation caused non-uniformity of local blur kernels can be constrained by reducing patch size, the spatial non-uniformity caused by translation should also be taken into consideration. Geometrically, the translation blur is depth dependent, so we favor the blur kernel estimated from a patch with consistent depth. Here we use reciprocal of standard deviation of depth in the patch as metric of depth consistency:

$$C_d(\text{patch}) = 1 / \sqrt{\sum_{(u,v) \in \text{patch}} (D(u,v) - \overline{D(u,v)})^2}, \quad (12)$$

where $D(u,v)$ is the corresponding depth at pixel (u,v) and $\overline{D(u,v)}$ is the mean depth in patch.

Considering above two criteria, we define our confidence of the estimation accuracy as:

$$C(\text{patch}) = C_d(\text{patch})C_H(\text{patch}). \quad (13)$$

To reduce computational cost, the blocks with low confidence weights are removed by setting the truncation threshold adaptively to preserve at least 95% of the total confidences.

B. PMDF Estimation

Theoretically, PMDF should be a continuous function, while in our paper we presented it by a set of uniformly distributed samples in the limited motion parameter space. For an arbitrary motion vector \mathbf{M} , the $p_{\mathbf{M}}(\mathbf{M})$ can be calculated by interpolation from discrete samples. Firstly, we perform parameterization to form the sample set in high-dimensional motion parameter space, and then vote the mappings of local blur kernels in high-dimensional motion parameter space to compute PMDF.

1) *Parameterization*: Suppose N is the limit of 2D kernel size, which can be set according to the blur level of image, the projection of high-dimensional camera motion in 2D kernel space should be no larger than N . Approximately, we could set the parameter range according to the following inequalities:

$$\begin{cases} \left| \frac{fT_{x,y}}{D_{min}} \right| < \frac{N}{2}, \left| \frac{T_z}{D_{min}} \sqrt{u_0^2 + v_0^2} \right| < \frac{\sqrt{2}N}{2}, \\ |f\theta_{x,y}| < \frac{N}{2}, \left| \sqrt{u_0^2 + v_0^2} \theta_z \right| < \frac{\sqrt{2}N}{2}, \end{cases} \quad (14)$$

where D_{min} is the minimum scene depth, f is focal length, and $\{u_0, v_0\}$ is the principal point coordinate of camera and in this paper which is set at the image center. Note that, f, u_0, v_0 are all in terms of pixel, while $D_{min}, T_{x,y,z}, \theta_{x,y,z}$ are all in terms of real units dimension.

To ensure estimation accuracy of PMDF, the 2D projection in imaging plane of each discrete level should be no larger than one pixel to eliminate ambiguity, consequently the number of discretization levels along each axis should be more than N . When the camera motion during exposure is slight, uniformly discretizing each parameter will lead to a near uniform sampling in 2D blur kernel space, i.e. the interval in 2D blur kernel domain between projection of two adjacent samples is approximately one pixel as well. To be on the safe side, we set $1.2N$ discretization levels along each dimension in this paper.

2) *Back-Projection*: As defined in Eq. 2, the 2D local blur kernels can be regarded as the 2D projection of 6D camera motion. Inversely, for a certain 2D local blur kernel, there are a set of samples meeting the projection. The so-called back-projection step is trying to calculate this set and their corresponding weights for a certain local blur kernel.

Given the patch center \mathbf{x} and depth $D(\mathbf{x})$, the 2D projection point $\mathbf{x}'(\mathbf{M}_i)$ of sample \mathbf{M}_i can be calculated by Eq. 2. For back-projection, we traverse the 6D motion parameter space, and set each sample the same weight as its 2D projection \mathbf{x}' in 2D local blur kernel K :

$$\mathcal{K}(\mathbf{M}_i) = K(\mathbf{x}'(\mathbf{M}_i)), \quad (15)$$

where i is the index of samples and traverses all the N^6 samples in the limited high-dimensional motion parameter space and \mathcal{K} is the 6D mapping of 2D local blur kernel.

3) *Robust Voting*: In this step, the PMDF is computed by a weighted voting process. For each sample in high-dimensional parameter space, its probability density can be estimated by

direct voting from J patches:

$$\mathbf{Q}^{K_1, K_2, \dots, K_J}(\mathbf{M}_i) \propto \sum_{j=1}^J \mathcal{K}_j(\mathbf{M}_i), \quad (16)$$

where j is the index of image patches.

Ideally, $\mathbf{Q}^{K_1, K_2, \dots, K_J}(\mathbf{M}_i)$ can be directly used as PMDF. However, considering the contamination from bad hypotheses (i.e. outliers of 2D local blur kernels estimated by uniform method), we propose a robust voting method to compute PMDF. As is well known, median filter is widely used in robust estimation for its desirable ability in suppressing the effects from bad hypotheses, but it cannot deal with the white noise with short-tailed distribution. In contrast, the Gaussian filter achieves promising performance for short-tailed noise, but suffers from the outliers. Therefore, we adopt order-based bilateral weighted voting which combines median filter and Gaussian filter to achieve good performance under both bad hypotheses and short-tailed noise of good hypotheses.

$$\begin{aligned} \mathbf{Q}(\mathbf{M}_i) &\propto \frac{\sum_{j=1}^J w_j(\mathbf{M}_i) C(patch_j) \mathcal{K}_j(\mathbf{M}_i)}{\sum_{j=1}^J w_j(\mathbf{M}_i) C(patch_j)} \\ w_j(\mathbf{M}_i) &= \exp\left\{-\frac{||\mathcal{K}_j(\mathbf{M}_i) - \text{Med}_j(\mathcal{K}_j(\mathbf{M}_i))||}{2\sigma_r^2}\right\}. \end{aligned} \quad (17)$$

Here, $\text{Med}(\cdot)$ is median operator, σ_r is the standard deviation of Gaussian distribution. The value of σ_r depends on the noise distribution of the estimated local blur kernels and we set it to be 0.05 empirically in our experiments.

Fig. 3 shows all the intermediate results of PMDF computation on a synthetic example, which is generated by applying a random camera motion blur on the images of Venus dataset [43] with known ground truth depth map. In this experiment, we set camera focal length $f=500$ pixels, the maximum 2D kernel size $N = 12$ pixels, the patch size is 100×100 pixels, the overlap size between adjacent patches is 15 pixels. The 2D projection of ground truth blur kernels and the estimated local blur kernels are respectively shown in Fig. 3(d) and (e). The confidence map is shown in Fig. 3(f) by brightness, from which one can see that the blur kernels in Fig. 3(d) and (e) are mostly consistent in regions with high confidence, i.e., our confidence approximately reflects the accuracy of estimated local blur kernels. Fig. 3(g)(h) give the estimated kernels by direct voting and confidence weighted robust voting respectively. For visualization, the resulted 6D PMDFs are projected to 2D blur kernel domain here. Fig. 3(i) gives close-up views of Fig. 3(d)(e)(g)(h) respectively. We can see that our confidence weighted robust voting framework can achieve good performance even when estimated local blur kernels are badly contaminated by outliers (as shown in the top-right region of Fig. 3(e)). The estimated PMDF fits the spatial-varying blur well over the whole image area, and thus can help the motion parameters converge to the right solution.

IV. PMDF-GUIDED NON-UNIFORM DEBLURRING

Utilizing the PMDF to guide our non-uniform deblurring, we can infer the latent image by Maximum A Posteriori (MAP).

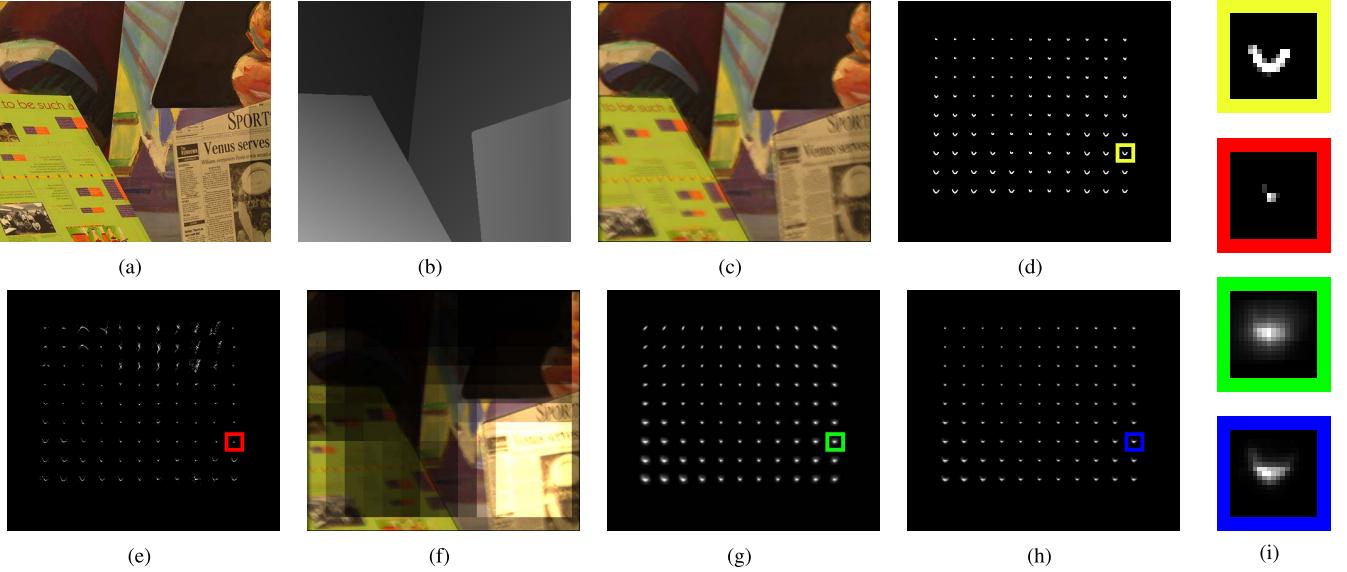


Fig. 3. Removing synthetic motion blur. (a) The latent sharp image. (b) The depth map. (c) The blurred image. (d) The ground truth local blur kernels. (e) The estimated local blur kernels. (f) The weighted confidence map which will be used in the back-projection step. (g),(h) The reprojections of PMDF back to 2D blur kernel space. (g) shows the directly voting result by Eq. 16, (h) is the robust voting result by Eq. 17. (i) gives details of (d)(e)(g)(h) respectively.

A. Objective Function

Following the Bayesian theorem, the joint conditional probability $p(\mathbf{L}, \mathbf{M}|\mathbf{I})$ can be decomposed as:

$$p(\mathbf{L}, \mathbf{M}|\mathbf{I}) \propto p(\mathbf{I}|\mathbf{L}, \mathbf{M})p(\mathbf{L})p(\mathbf{M}). \quad (18)$$

Here, $p(\mathbf{I}|\mathbf{L}, \mathbf{M})$ denotes the likelihood, and $p(\mathbf{L}), p(\mathbf{M})$ are the prior of latent image and camera motion respectively. In this paper, we use the PMDF $\mathbf{Q}(\mathbf{M})$ to replace the motion prior $p(\mathbf{M})$. The above function can be optimized by minimizing the following energy function:

$$E = \|\mathbf{I} - \hat{\mathbf{I}}\|^2 + \lambda_L \|\nabla \mathbf{L}\|^{0.8} - \lambda_M \sum_{t=0}^{N_t} \log \mathbf{Q}(\mathbf{M}_t), \quad (19)$$

where $\hat{\mathbf{I}}$ is the blurred image predicted from latent image \mathbf{L} and camera motion \mathbf{M} according to Eq. 6. The 2nd item is used to constrain the sparseness of the gradient of \mathbf{L} . The last item denotes the constraint of high-dimensional camera motion, which represents the consistency between current parameters and precomputed PMDF. λ_L and λ_M are weighty factors. In our experiments, λ_L is assigned with the same value in [44], and λ_M is initialized to be 0.03 and with a 50% decreasing rate in the following iteration.

B. Optimization

We minimize Eq. 19 by optimizing latent image \mathbf{L} and camera motion \mathbf{M} alternately. First, we initialize the camera motion \mathbf{M} by sampling from distribution of camera motion (i.e., PMDF), and initialize latent image \mathbf{L} as weighted stitching of deblurred patches estimated by [4]. The optimization is conducted by following two steps.

1) Fixing \mathbf{L} and Optimizing TSMF \mathbf{M}_t : Assuming the latent image \mathbf{L} be known, Eq. 19 can be simplified by eliminating terms which depend only on \mathbf{L} :

$$E = \|\mathbf{I} - \hat{\mathbf{I}}\|^2 - \lambda_M \sum_{t=0}^{N_t} \log \mathbf{Q}(\mathbf{M}_t). \quad (20)$$

For a pixel in the textureless regions, the predicted value $\hat{I}(u, v)$ will not deviate from the observed value $I(u, v)$ too much even the camera motion estimation is not accurate. Therefore, these terms can be discarded in optimizing camera motion step. Then, Eq. 20 becomes:

$$E = \sum_{u,v} \Omega(\nabla \mathbf{I}, u, v, \frac{N}{2}, \sigma) (I(u, v) - \hat{I}(u, v))^2 - \lambda_M \sum_{t=0}^{N_t} \log \mathbf{Q}(\mathbf{M}_t), \quad (21)$$

where $\Omega(\cdot)$ is the indicate function defined by:

$$\Omega(\nabla \mathbf{I}, u, v, \frac{N}{2}, \sigma) = \begin{cases} 1 & \text{if } \exists(m, n) \quad \|\nabla I(m, n)\|_2 > \sigma, \\ & \forall(m, n) \in \{(m, n) | (m-u)^2 + (n-v)^2 < (\frac{N}{2})^2\} \\ 0 & \text{otherwise} \end{cases} \quad (22)$$

Here N is size limit of 2D blur kernel, σ is the threshold for choosing pixels with significant gradient. White area in Fig. 4(a) shows the selected regions.

All of the variables in Eq. 21 are continuous, and can be optimized by the traditional gradient based optimization methods. In this paper, Levenberg-Marquard algorithm is used to minimize Eq. 21. Adding smooth constraint to camera trajectory \mathbf{M}_t is beneficial for its computation, but tends to

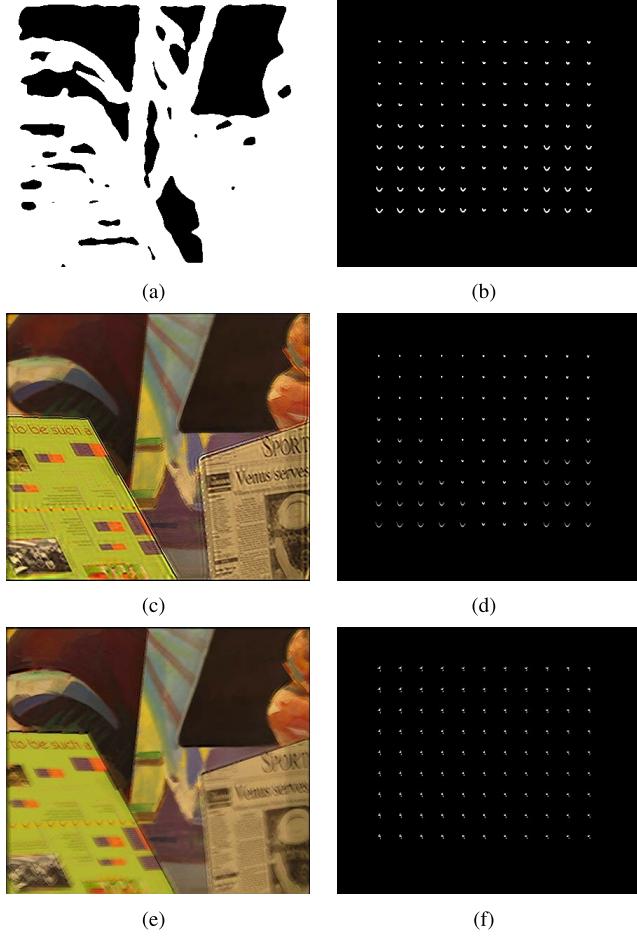


Fig. 4. Result of our PMDF guided deblurring method. (a) Mask for covering textureless region. (b) The groundtruth local blur kernels. (c) Final deblurring result. (d) The local blur kernels. (e) The result by Whyte *et al.* [22]. (f) The local blur kernels computed by Whyte *et al.* [22].

complex the optimization and may hamper the convergence. Therefore, we directly smooth it using Gaussian filter after each iteration instead of adding an additional regularizer into the objective function.

In the first iteration, the latent image \mathbf{I} can be initialized by stitching deblurred patches. To suppress the bad deblurred results, the confidence map (shown in Fig. 3(f)) is used to weight each pixel in Eq. 20, then the optimizing function becomes:

$$E = \sum_{u,v} \Omega(\nabla\mathbf{I}, u, v, \frac{N}{2}, \sigma) w_c(u, v) (I(u, v) - \hat{\mathbf{L}}_0(u, v))^2 + \lambda_M \sum_{t=0}^{N_t} \log \mathbf{Q}(\mathbf{M}_t), \quad (23)$$

where $\forall i \in \{i | (u, v) \in patch_i\}$,

$$w_c(u, v) = \overline{C(patch_i)} \\ \mathbf{L}_0(u, v) = \frac{\sum C(patch_i) patch_i(u, v)}{\sum C(patch_i)} \quad (24)$$

Here, $C(patch_i)$ is the confidence value. Since overlaps exist between adjacent patches, the mean value $\overline{(\cdot)}$ is introduced.

Similarly the pixel intensity $\mathbf{L}_0(u, v)$ is computed by weighted averaging for initialization.

2) *Fixing TSMF \mathbf{M}_t and Optimizing \mathbf{L} :* Eliminating all the terms depending on \mathbf{M} , Eq. 19 can be simplified as:

$$E = ||\mathbf{I} - \hat{\mathbf{I}}||^2 + \lambda_L ||\nabla\mathbf{L}||^{0.8}. \quad (25)$$

This is a typical non-blind spatially varying deblurring problem, which can be solved by using iterative re-weighted least-squares methods [45], [46], which replace all the convolution operations in uniform algorithms with our pixel-wise blurring process defined in Eq. 6.

Fig. 4 gives the deblurring result by using our PMDF guided method. It is obvious that the result derived by our method could achieve large improvement over Whyte *et al.*'s which considers only 3 rotation DoFs.

V. ANALYSIS AND EXPERIMENTS

In this section, experiments are conducted to analyze the model parameters and validate the proposed algorithm.

A. Manifold Analysis

To illustrate the effectiveness of our framework and show the principle of PMDF intuitively, we analyze the characteristics of mapping from 2D blur kernel space to high-dimensional parameter space.

For easier visualization, we constrict the degrees of camera motion to be 3. To make the experiments more representative, we choose the translation along x -, y - axes (T_x, T_y) and rotation around z - axes R_z , which are typical camera motion and widely used in previous works [27] and [21].

1) *Impulse Kernel:* As is well known, a blurred image can be regarded as the integration of a series of images warped from the latent image. Each warped image corresponds to a camera pose which is located on the trajectory of the camera motion. For each pixel \mathbf{x} in the latent image, the location of its corresponding pixel \mathbf{x}' in warped image can be computed by Eq. 2 and 3 with the depth $D(\mathbf{x})$.

Considering the 2D blur kernel domain, since 2D kernels are projections of camera motion, it can be regarded as a summation of a series of impulse kernels which correspond to the camera poses along the camera motion trajectory. The center of the impulse kernel locates at the pixel \mathbf{x} and the peak value of the kernel has an offset $d\mathbf{x}$, which is used to transform the pixel \mathbf{x} on latent image to its corresponding pixel \mathbf{x}' on warped image, and we denote the offset as $d\mathbf{x} = \mathbf{x}' - \mathbf{x}$.

In the case of 2D impulse kernel, the peak point corresponds to a $N - 2$ dimensional manifold in N dimensional parameter space. Specific to the 3D parameter space case, this manifold is 1D and can be embedded into 3D Euclidean space. We will show the mapping manifolds of these impulse kernels and analyze the effects of three variables \mathbf{x} , $D(\mathbf{x})$ and $d\mathbf{x}$ on these manifolds respectively.

a) *Kernel center \mathbf{x} :* Kernel center \mathbf{x} describes the location of the 2D local blur kernel and is also the center of corresponding pixel in latent image. Here we fix $d\mathbf{x}$ to be $(0, 0)$ and $D(\mathbf{x})$ to be 5m, and change \mathbf{x} in both horizontal and vertical directions. The mapping manifolds in 3D parameter space

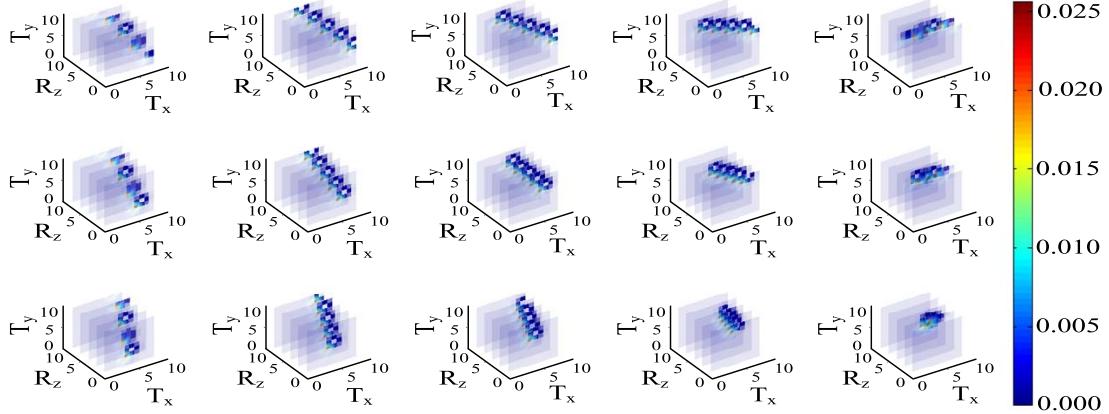


Fig. 5. 1D mapping manifolds at different image locations. The horizontal coordinates u (pixels) of columns and vertical coordinates v (pixels) of rows are labeled at the left and bottom of the figure.

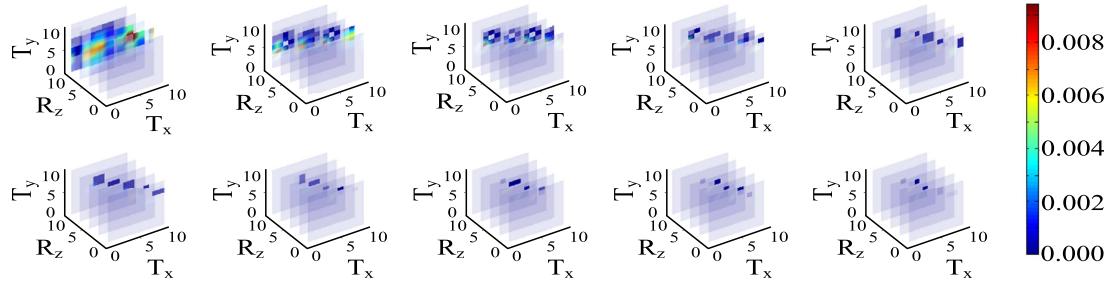


Fig. 6. 1D mapping manifolds at different depths. The depth D is labeled at the bottom of each subfigure.

are shown in Fig. 5, which reveals that these 1D manifolds are almost straight lines when motion parameters vary with a small range around origin, while the slopes of these lines change as the kernel center \mathbf{x} changes. However, all these manifolds intersect at the origin point which is the exact 3D motion parameters of camera pose coincide with these impulse kernels. In fact, for all the local impulse kernels corresponding to the same camera pose with different center \mathbf{x} , *the corresponding 1D manifolds cross through the point which exactly represents the camera pose, and have different slopes*.

b) *Depth $D(\mathbf{x})$* : Fig. 6 depicts the mapping manifolds with different depths, with the kernel center \mathbf{x} and offset $d\mathbf{x}$ being set constant. According to the figure, *the slopes of these 1D manifolds change with scene depth*. However, unlike \mathbf{x} , their thicknesses also vary with depth. This is in accordance with the projective geometry, i.e. the farther the scene point is, the more translation is needed to cause the same offset in 2D blur kernel space, an extreme case is when the depth is infinite translation will never cause any movement of the projection. Therefore, a pixel with the same size in blur kernel domain corresponds to a thicker manifold in parameter space as the depth increases. As the scene point moved nearer, the projection range of the manifold on translation axes expands while that on rotation axes shrinks. We can conclude that large depth helps to determine rotation, and small depth helps to determine translation, and *the scene with large depth range is in favor for high-dimensional camera motion estimation*.

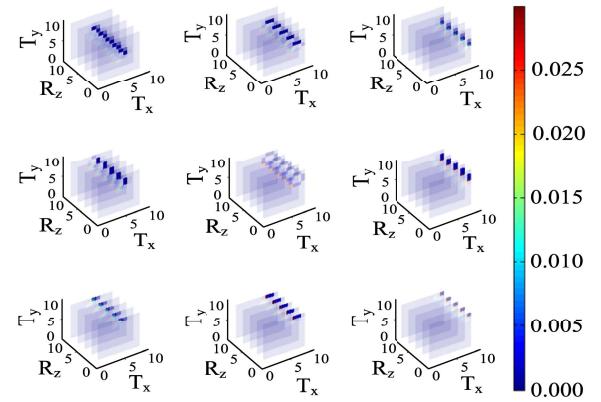


Fig. 7. 1D mapping manifolds with different offsets. The horizontal and vertical offset du (pixels), dv (pixels) of columns and rows are labeled at the left and bottom of the figure.

c) *Offset $d\mathbf{x}$* : Offset $d\mathbf{x}$ describes the shift of peak point of impulse kernel. As we can see in Fig. 7, for a certain scene point (i.e. the \mathbf{x} and $D(\mathbf{x})$ are constant) *the varying of $d\mathbf{x}$ corresponds to translation of the 1D mapping manifold on T_x , T_y directions in 3D parameter space*.

2) *Continuous Kernel*: One important property of motion blur kernel is the continuity, so that the kernel can be regarded as a trajectory of a moving speckle, with the velocity reflected by the intensity of pixels along the trajectory. As discussed above, the varying of offset $d\mathbf{x}$ corresponds to translation of

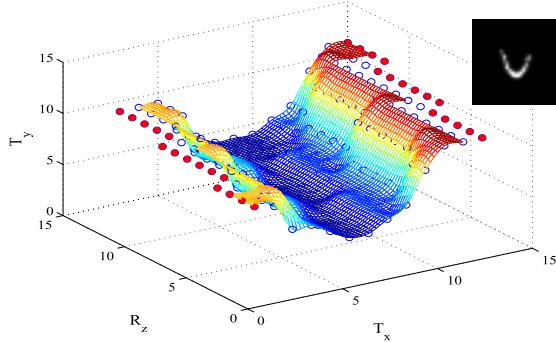


Fig. 8. 2D mapping manifold of a curve-like kernel. The corresponding 2D blur kernel is shown at the top right of the figure.

mapping manifold along T_x, T_y directions in 3D parameter space, so the 1D manifold spans to be a 2D surface in 3D parameter space, as shown in Fig. 8. The points with the same color in the surface have the same probability motion density. We set \mathbf{x} to be $(0, 0)$ and $D(\mathbf{x})$ to be 5m and give the surface manifold of the continuous kernel in 3D motion parameter space. The unknown camera motion trajectory is a curve along the surface from one side to the other (both sides are marked with small filled circles in Fig. 8). Furthermore, local blur kernels estimated from different kernel center and scene depths lead to different mapping surfaces, their intersection determines the 3D camera motion trajectory.

3) Parameter Space With Dimension Higher Than 3: Although the above discussions are all based on the 3D parameter space, the conclusions can be extended to higher-dimensional space: (i) 2D blur kernels estimated from each patch are projected from the common camera motion, so the intersection of their projection in high-dimensional space fits the camera motion trajectory; (ii) big scene depth (i.e. farther scene) helps to determine camera rotation parameters, while small scene depth (i.e. nearer scene) helps to determine translation. Thus, large depth range scene helps to determine the unknown camera motion, and small depth range may lead to model degeneration, i.e. the problem has multi-solutions. (iii) incremental shift between latent image and blurred image reflects the movement of mapping manifold along translation directions (x -, y - axes).

B. Analysis of Depth-Dependent Blur

Here, we analyze some characteristics of depth-dependent blur caused by camera translation.

1) Sensitivity to Sudden Depth Change: Both translation and rotation may cause non-uniform blur, however, the rotation blur varies gradually over the image, so it is possible to find a single PSF with sufficient accuracy for a small local region, and obtain promising deblurring results by deconvolution.

In contrast, the translation blur changes significantly at the step edges that reflect discontinuous scene depth, and ignoring the abrupt change of PSFs at these positions would deteriorate the deblurring performance by introducing or aggravating ringing artifacts. Therefore, compared with the rotation, the blur caused by translation is more challenging in image deblurring.

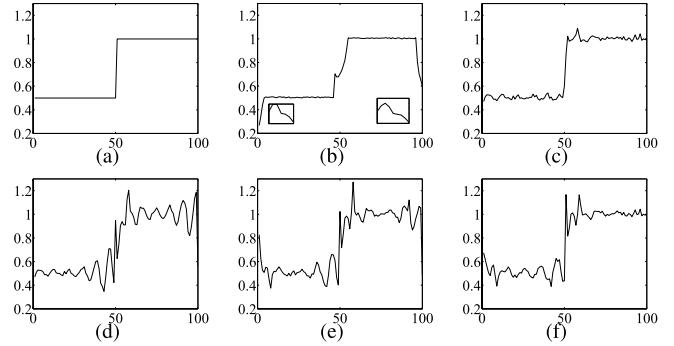


Fig. 9. Deblurring of 1D step signal with blur kernels with abrupt size changes. (a) shows the original 1D signal. (b) shows the blurring result of signal in (a) by two different sized kernels shown under the curve. Before the step, it is blurred by the smaller kernel (7 units), while after the step, the larger kernel (9 units) is used. In addition, the synthetic blur signal is contaminated by Gaussian noise with $\sigma = 0.01$. (c) The signal retrieved by non-uniform RL deblurring method. (d–f) The signal retrieved by uniform RL method with 7, 8, 9 units blur kernels respectively.

Here we show an example of 1D step signal (see Fig. 9(a)) blurred by kernels suddenly scaling at the step point, as shown in Fig. 9(b). This is quite similar to the situation of translation cause blur. Fig. 9(c–f) shows the result of non-uniform and uniform Richardson-Lucy (RL) [6] methods respectively. From the results by RL methods in Fig. 9(d–f), one can see that *inaccurate approximation of blur kernel will result in strong ringing artifacts near the stair*.

2) Over-Deblurring & Under-Deblurring: Although translation blur varies with scene depth, the general shape of the PSFs is preserved [24]. Therefore, inaccurate estimation of PSFs due to translation may result in two possible undesired outcomes: over-deblurring and under-deblurring, i.e., deblurring using a kernel larger and smaller than the actual size.

Fig. 10 shows an example on synthetic data. We translate camera along x -axis with constant velocity to blur the sharp image in Fig. 10(a) and generate (b). Fig. 10(c) and (d) are the ground-truth depth map and PSFs respectively. The PSFs show that translation blur is drastically non-uniform with depth varying. Fig. 10(e) gives the result of non-uniform RL which achieves good performance in both far and near regions. From the uniform deblurring results in (f–h) with increasing kernel sizes, we can conclude that the smaller blur kernel leads to incomplete deblurring, while the larger one introduces apparent ringing artifacts. Therefore, *depth-dependent blur kernel must be considered* to remove the non-uniform blur while suppress the ringing artifacts along depth edges.

C. Degeneration Cases

Although, varying depth may causes significant non-uniform blur, it can be ignored in some special cases. In such cases, the non-uniformity of image blur is almost independent of the scene depth. We can reduce the DoFs of blur model in specific conditions to reduce the computation cost. Our method provides a flexible framework for setting the DoFs of camera motion, i.e. user can use a subspace of the entire 6D parameter space as the support field of PMDF. This section gives experiment results in some typical degeneration cases.

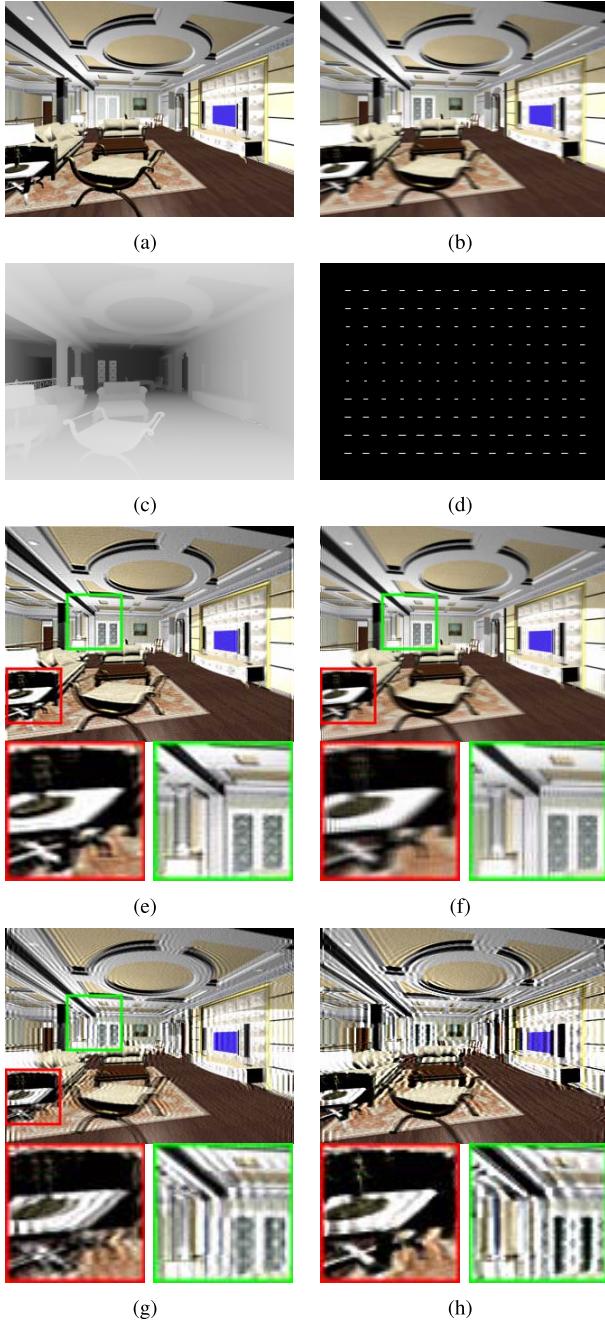


Fig. 10. Synthetic blur caused by translation. (a) A sharp image of a large depth-range scene. (b) The synthesized blurred image caused by camera translation along x -axis. (c) The ground truth depth of the scene. (d) The spatially varying PSFs. (e) The restored image derived by non-uniform RL deblurring method. (f–h) The restored image by uniform RL deblurring method with blur kernel length being 2, 5, 8 pixels respectively.

1) Narrow Depth Range Scene: If the scene has a narrow depth range, the projective blur model degenerates from perspective transform to homography transform, then scene depth can be ignored since it would not cause spatially varying blur. In this case, 4D subspace with 3 translations and the in-plane rotation could fit all the local blur kernels very well.

Fig. 11 shows the degeneration in case of a planar scene. In this simulated experiment, we set all the scene points with a constant depth and generate a 6D camera motion

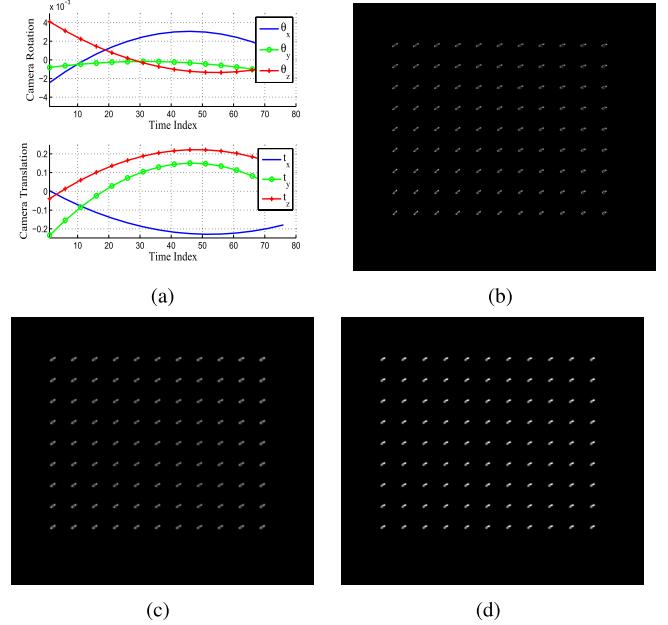


Fig. 11. Model degeneration in case of planar scene. (a) Synthetic 6-DoF camera motion, the unit of rotation is angle degree and that of translation is times of focal length. (b) Groundtruth local blur kernels. (c) 2D projection of PMDF with 4 DoFs. (d) 2D projection of PMDF with 6 DoFs.

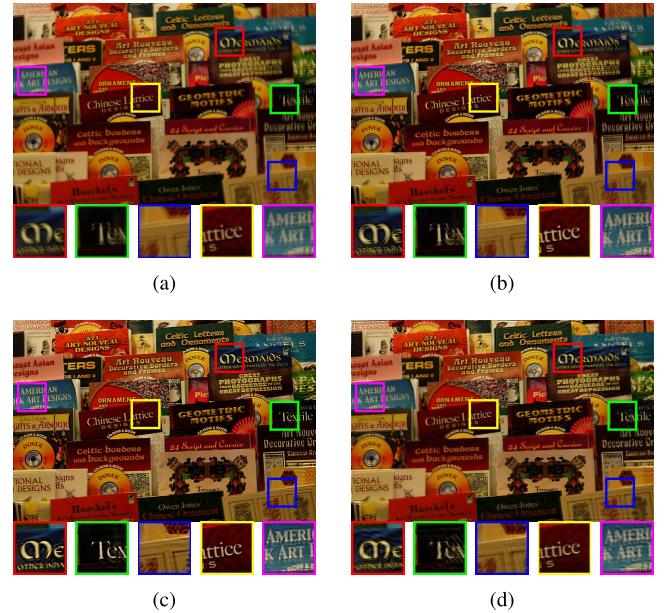


Fig. 12. Deblurring results of a planar scene. Parameter settings: focal length f is assumed to be 1000 pixels, scene locates at a constant depth $d=50 \times f$, maximum 2D blur kernel size $N=15$ pixels, patch size is 150×150 pixels and overlap size is 20. (a) Observed blurred image. (b)(c) The spatially invariant and varying deblurred result shown in Gupta et al. [27]'s uniform deblurring method. (d) The result of our method.

randomly (shown in Fig. 11(a)). The local blur kernels are shown in Fig. 11(b). The 6D and 4D (3 translation and in-plane rotation) parameter space were used to estimate PMDFs separately. To show the performance of the estimated PMDFs, we project the PMDFs back to 2D kernel space for comparison. From the 2D projection results in Fig. 11(c)(d),

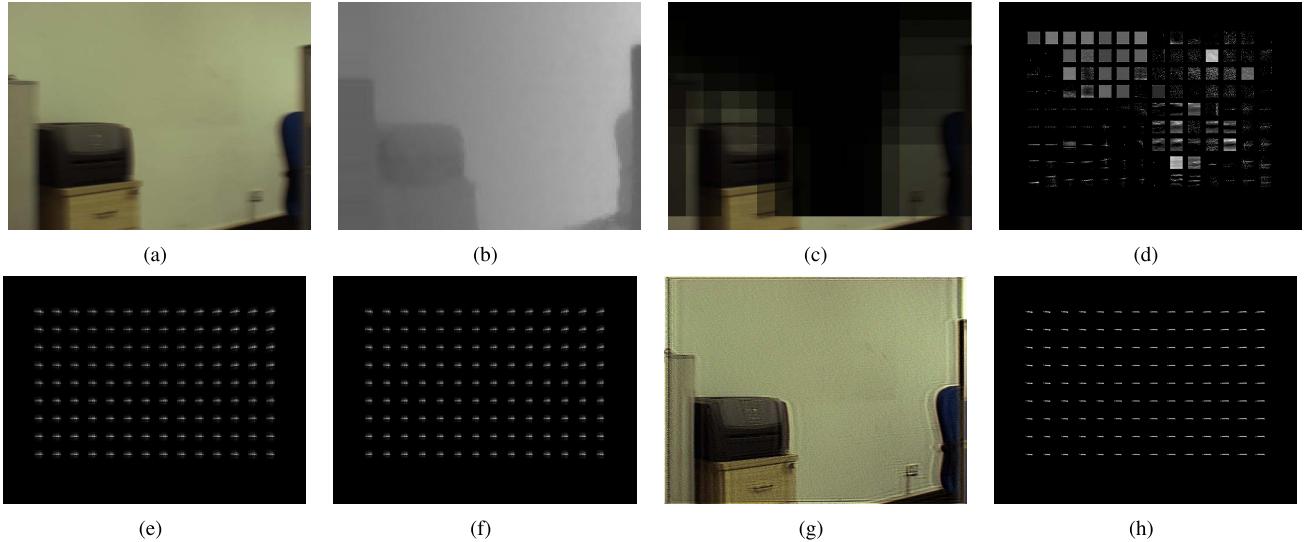


Fig. 13. Deblurring result on scenes lacking of credible local blur kernels. Camera focal length $f=440$ pixels, 2D kernel size N is set to be 20 pixels, patch size is 120×120 pixels and overlap size is 25 pixels. (a) Blurred image. (b) Depth map captured by depth camera. (c) Confidence map. (d) Estimated local blur kernels. (e) 2D projection of PMDF with 3-DoF. (f) 2D projection of PMDF with 6-DoF. (g) Deblurred image. (h) Local blur kernels computed by our method.

it is obvious that both 6D and 4D parameter spaces fit all the blur kernels well. However, the PMDF estimator with 4D subspace is significantly faster than that with whole 6D parameter space.

We perform the deblurring under 4D DoFs assumption and compare the deblurring result with that of spatially invariant and varying algorithm in Gupta *et al.* [27], as shown in Fig. 12(b) and (c). The results reveal that, the uniform model fails to recover sufficient details, while both Gupta *et al.*'s model and ours can describe the non-uniform blur in planar scene and give promising deblurring results.

Comparing two non-uniform methods, our recovery is slightly inferior to that of [27]. In fact, this slight inferiority is reasonable and a similar phenomena is also observed by Köhler *et al.* in [47]. In their benchmarking, Cho and Lee's uniform deblurring method [42] achieves better results than Hirsch *et al.*'s [29] for some small spatial variance blurring cases. These two methods adopt a similar framework, while the latter one uses some strategies (e.g. EFF) for 3D non-uniform cases. However, according to the Occam's Razor theory, a complex model may work worse than the simpler ones for the degeneration cases.

2) Lacking Credible Local Blur Kernels: Although most blurred image can provide enough information to estimate local blur kernels, but not all of them are credible enough to constrain the PMDF. For instance, if a scene includes a large textureless region, the local blur kernels estimated from which are awful and all the credible blur kernels concentrate on a local area with a nearly constant depth map. In this case, there is no sufficient credible local blur kernels to determine the exact camera motion, so we can just reduce the dimension of parameter space and ignore the scene depth.

Fig. 13 shows an example of this case. Since the blurred image has large textureless background and only the kernels located in the lower left corner are credible, as

shown in Fig. 13(c)(d) respectively. The blurred/depth image pair is captured by RGB/depth hybrid acquisition system which combines Point Grey FL2-08S2C and SwissRanger SR4000. The depth camera is applicable for scenes within 10m and suffices in most indoor cases. The camera system is calibrated using the method in [39]. The RGB image and its corresponding depth captured by our hybrid camera are shown in Fig. 13(a)(b). Fig. 13(e)(f) show the 2D projection of PMDF with 6D and 3D (3 Rotations) parameter space, and Fig. 13(g)(h) give the deblurred result and final kernels estimated by reducing Dof to 3. There also exist some ringing artifacts along the depth edges, this is due to rich texture locates mostly in the bottom left corner and this part is coarsely of a uniform depth, thus the illposedness of 6D PSF estimation is more severe. In spite of these imperfections, our algorithm still recovers most regions reasonably for such extreme cases.

3) Distant Scene: In this experiment, we show the degeneration caused by distant scene. Fig. 14 shows the example obtained from [22], whose capture settings are provided on the website: the focal length of camera is 35.1mm, CCD size is $36\text{mm} \times 24\text{mm}$, and the resolution is 259×194 pixels. The depth information is not provided, but it can be deduced by common sense that the distance from camera to scene is more than 20m, then we can compute that at least 78.9mm translation cause a pixel width blur. Therefore, the translation can be ignored in this example.

We perform the deblurring with 3 rotation DoFs assumption and compare the deblurring result with that of Fergus *et al.* [4] and Whyte *et al.* [22], as shown in (b)(c). The results reveal that, both Whyte *et al.*'s method and ours can give promising deblurring results, while the Fergus *et al.*'s fails to recover high frequency details. This is due to the fact that the blur caused by in-plane rotation is intrinsically non-uniform, thus the uniform deblur approach is infeasible for this example.

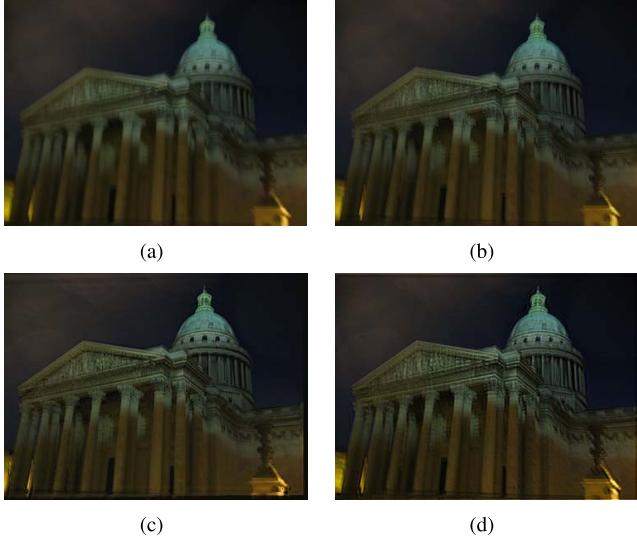


Fig. 14. Deblurring results of distant scene. Parameter settings: focal length $f=252$ pixels, maximum 2D kernel size $N=20$ pixels, patch size is set to be 100×100 pixels and overlap size is 25 pixels. (a) Observed blurred image. (b) The result of Fergus *et al.* [4]'s uniform deblurring method. (c) The result of Whyte *et al.* [22]'s methods. (d) The result of our method.

In all, our approach is apparently superior to the uniform methods, while generally comparable to state-of-the-art non-uniform ones in degeneration cases, although being slightly inferior (in some regions) to the non-uniform methods built specially for the degeneration cases. The inferiority is mainly due to following reasons: 1) We adopt some strategies (e.g. TSMF) to decrease computational cost and thus can deal with high dimensional camera motion. These strategies may harm the performance a bit, but our algorithm still gives promising results. 2) Occam's rule implies that a more complex model may work worse for degenerated cases. 3) The robustness of algorithm suffers from the errors and noise in kernel estimation.

D. Experiment Results With Given Depth Map

1) Image Deblurring With Synthetic Depth and Motion:

We select an image with large depth range and blur it by a simulated 6D camera motion and intrinsics. The blurred image is shown in Fig. 15(a), while the sharp version and corresponding depth map have been shown in Fig. 3(a)(b). Former analysis shows that the PSFs are mainly affected by rotation in farther regions, while the translation component dominates in nearer regions, as shown in Fig. 15(b). For comparison, we show the results of Cho and Lee's [42], Whyte *et al.*'s [22], Hirsch *et al.*'s [29] and Xu *et al.* [25]'s methods in Fig. 15(c)(e)(f)(g) respectively in parallel with ours in Fig. 15(h). Besides, to show the effectiveness of the global constraint, we demonstrate the result of a naive fix-size-patch non-uniform deblurring algorithm with the same patch size as our method. For each patch, a state-of-the-art uniform method (Xu *et al.* [9]) is applied. Considering that the size of the blurred image is relatively small (383×434 pixels) compared to the patch size (100×100 pixels), this experiment is very challenging for our patch based framework. Obviously, our approach gives promising results in all the regions while the

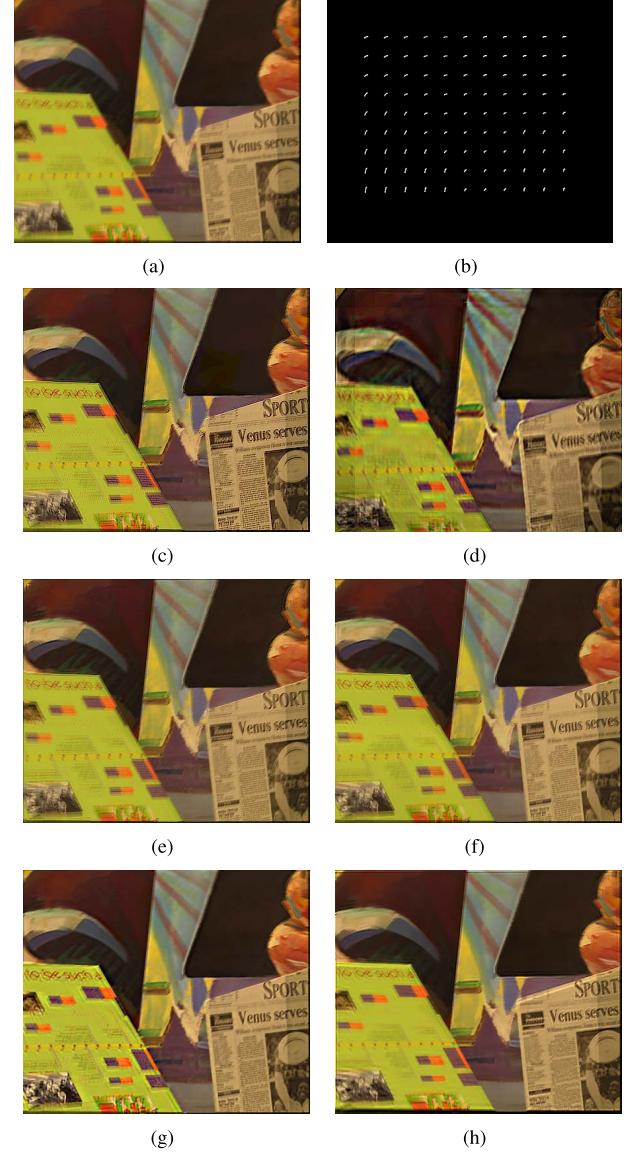


Fig. 15. Deblur results on synthetic data. Parameter settings: focal length $f=500$ pixels, 2D kernel size $N=12$ pixels, patch size is 100×100 and overlap size is 15 pixels. (a) Blurred image. (b) Ground truth PSFs map. (c) Result by Cho and Lee [42]. (d) Result by naive patch-based deblurring without global constraints. (e) Result by Whyte *et al.* [22]. (f) Result by Hirsch *et al.* [29]. (g) Result by Xu *et al.* [25]. (h) Our result.

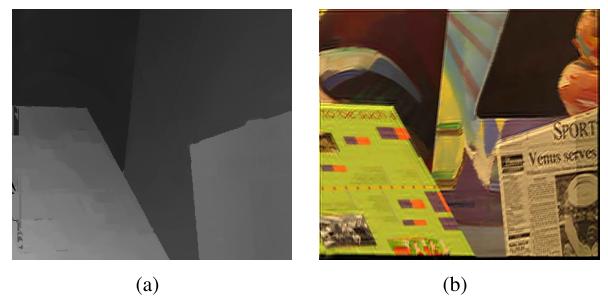


Fig. 16. Deblur results with inaccurate depth map. Result with depth computed from multiview images by stereo method [48].

other methods fail in some regions, because neither depth is considered nor the adopted motion model can approximate the 6D camera motion well enough.

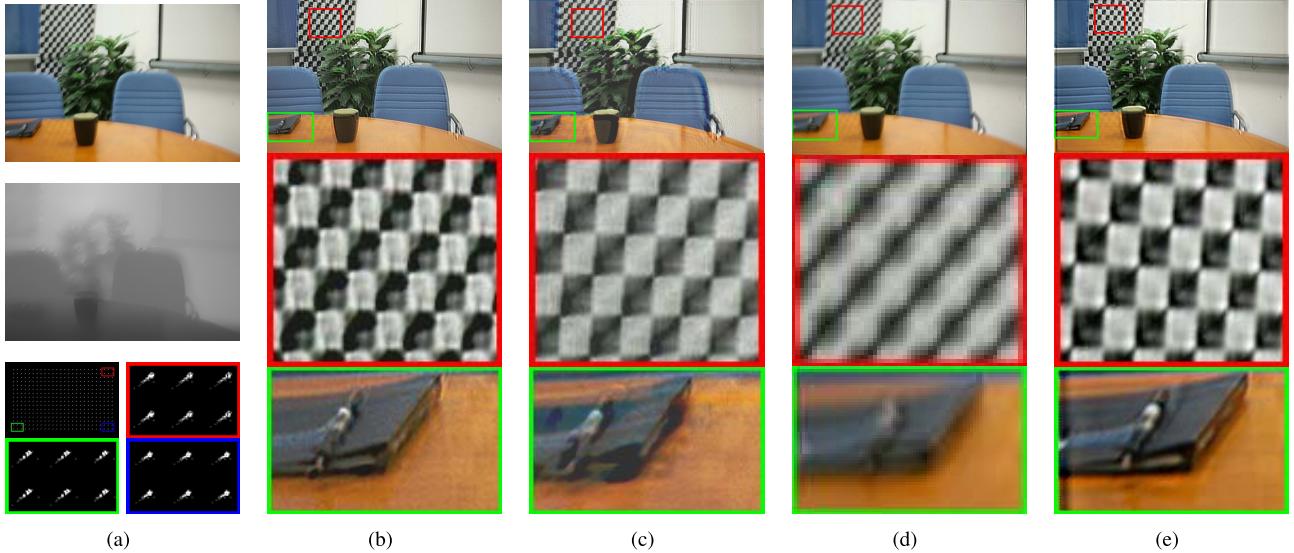


Fig. 17. Deblur results on a blurred/depth image pair captured by hybrid camera system. Parameter settings: The RGB camera's focal length $f=751$ pixels, 2D blur kernel size $N=20$ pixels, patch size is 180×180 pixels and overlap size is 30 pixels. (a) Input and intermediate results. (b)(c) Results of Fergus *et al.* [4]’s algorithm by choosing areas marked by green/red panes for kernel estimation respectively. (d) Result of Whyte *et al.* [22]’s non-uniform method. (e) Our result.

To test the tolerance of the proposed method to depth inaccuracy, we applied our algorithm on the depth estimated from the multiviews of Venus dataset [43] by Min and Kwanghoon’s method [48], as shown in Fig. 16. One can notice that the performance is slightly affected but still quite promising compared to that given true depth in Fig. 15(h). The robustness to depth inaccuracy is mainly due to the proposed weighted voting strategy.

2) Image Deblurring With Depth Camera Assistance: To display the integral performance of the whole framework, we test our deblurring algorithm on data captured by a RGB/depth hybrid acquisition system. The blurred image and depth are shown in the top and middle image in Fig. 17(a). From the final estimated local blur kernel map (shown in bottom of Fig. 17(a)), we can see that the blur kernels are spatially varying. Fig. 17(b)(c)(d)(e) give comparison among deblurring results of Fergus *et al.*’s [4], Whyte *et al.*’s [22] and ours, with the close-ups shown below. Fig. 17(b)(c) respectively give results of Fergus *et al.*’s [4] by selecting red and green region for blur kernel estimation. It is obvious that the uniform algorithms perform well in the selected region while the performances in other areas deteriorate. On the contrary, we achieve good performance over the whole image. In addition, the algorithm of Whyte *et al.* [22] works almost 20 times (1 hour vs. 20 hours) slower than ours under the same computing conditions (matlab code under Intel Core 2.1GHz, 4G RAM).

3) Image Deblurring With Depth Computed by Structure From Motion Method: In practical applications, depth acquisition equipment may not be easily available for some scenarios. Fortunately, our methods does not need a high quality depth map, so we can use the depth map derived by structure from motion methods. To deal with the feature correspondence between blurred images, we downsample the original blurred

image to derive the coarse depth map and then smooth it by bilateral filtering after upsampling.

An image sequence with large depth range is captured from 8 different views, and introduce some arbitrary camera shakes manually. The first image in Fig. 18(a) shows the 2nd frame of the sequence, and the scene depth estimated via stereo method is in the middle image of column (a) (shown as disparity map), and computed local blur kernel at each pixel is shown in the bottom image in column (a). The deblur result and some detailed regions are displayed in Fig. 18(e), in parallel with that of Fergus *et al.* [4] in column (b), Shan *et al.* [4] in column (c) and Whyte *et al.* [22] in column (d) as well.

The blur kernel of Fergus *et al.*’s method [4] is estimated by (manually) selecting a local region of the blurred image. Therefore, their method achieves good result in this area (indicated by red rectangle), but serious ringing artifacts appear in background region. Differently, Shan *et al.* [8] estimate a global kernel from whole image, thus there exist apparent artifacts in both foreground and background areas. Meanwhile, Whyte *et al.* [22] ignores depth variation and camera translation and the result isn’t good enough either. In contrast, our method performs estimation explicitly to compute non-uniform blur kernels, and gives promising results in areas at different depths, as shown in Fig. 18(e). Our result still has a slight of artifacts along the occlusion boundaries (doll face and book ridges) caused by depth estimation errors.

4) Image Deblurring With Binocular Stereo: We also apply our algorithm on the Mickey data set provided by Xu *et al.* [25]. As shown in Fig. 19(e), the depth map is incorrect since it gives a constant value to several obviously slant surfaces. We estimated the 2D local blur kernels from patches (shown in Fig. 19(b)), and projected them back to 3D motion parameter space(x , y - translation and in-plane rotation) to compute the PMDF (shown in Fig. 19(c)). The result of

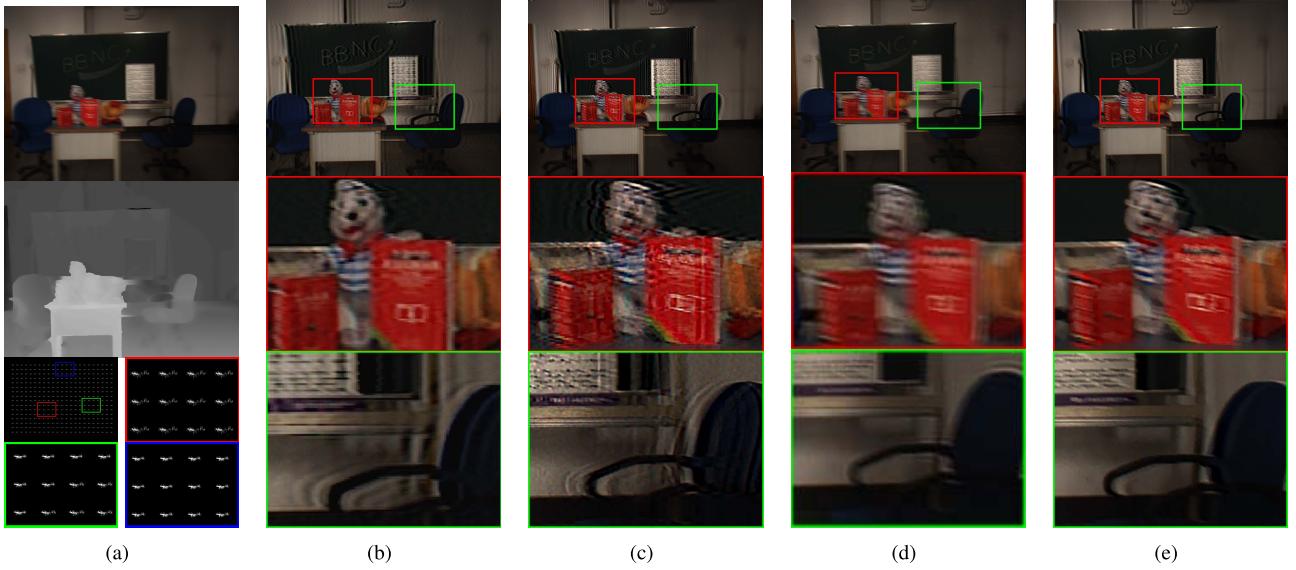


Fig. 18. Deblur results on a multiview image set. Parameter settings: The focal length of the cameras $f = 1761$ pixels, 2D kernel size $N=25$ pixels, patch size is set to 200×200 pixels and overlap size is 30 pixels. (a) Input and intermediate results. (b–e) Result by Fergus *et al.* [4], shan *et al.* [8], Whyte *et al.* [22] and our method.

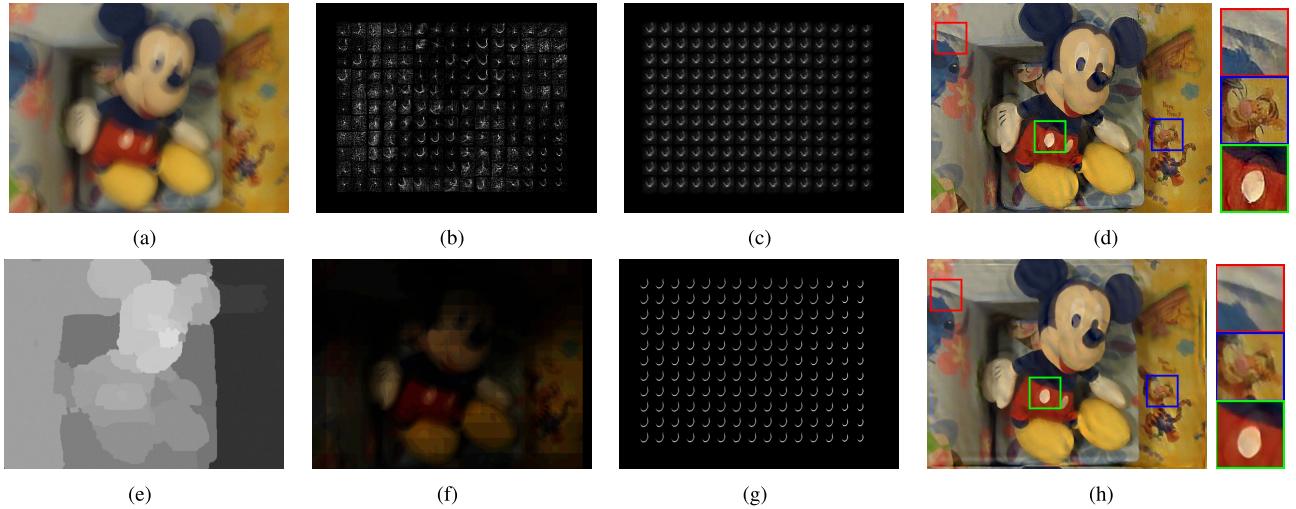


Fig. 19. Deblurring result on a scene with inaccurate depth. (a) Blurred image. (b) Initial local blur kernels. (c) 2D projections of PMDFs. (d) Xu *et al.* [25]'s deblur result. (e) Inaccurate depth map provided by Xu *et al.* [25]. (f) Confidence map. (g) Our final estimated blur kernels. (h) Our deblurring results.

our method is shown in Fig. 19(h), one can see that it gives reasonable performance in most regions. However, compared to Xu *et al.*'s [25] results, our method has more ringing artifacts. This is mainly because that the proposed approach requests that the blur kernels across the whole image lattice conform to one unified high-dimensional camera motion, and thus the performance is more sensitive to scene depth error, which occurs often along the occlusion boundaries.

VI. DISCUSSIONS AND FUTURE WORK

We have proposed a 6D depth-aware blur model derived from the projective geometry, and an effective camera motion estimation approach. Benefiting from the guidance of unified camera motion distribution estimated by local blur information, our method achieves satisfying deblurred result for

non-uniformly blurred images, especially in the situations with large depth range scene and apparent camera translation. For the degeneration situations, our framework can flexibly select the motion dimension to degrade the motion blur model, and thus can reduce computational cost and prevent over-fitting.

The robustness of deblurring methods is a problem concerned by the whole community, especially for such a complex model. This paper proposes backprojection and weighted voting strategy to raise the robustness to LBK estimation errors in some extreme cases. As shown in the experiments, our algorithm can give promising result when most estimated local blur kernels have intensive errors and noise, which shows the robustness of the proposed algorithm. However, the accuracy of locally estimated blur kernels is still crucial for our framework. In the future, we'll try to improve to make the algorithm more robust and adaptive.

In terms of computational speed, it can be greatly accelerated by GPU for most calculations within each iteration are highly parallel. In addition, out-of-focus effect which often co-occurs with camera shake blur under low light environment will also be considered as an extension.

REFERENCES

- [1] J. Cai, H. Ji, C. Liu, and Z. Shen, "Blind motion deblurring from a single image using sparse approximation," in *Proc. CVPR*, 2009, pp. 104–111.
- [2] X. Chen, X. He, J. Yang, and Q. Wu, "An effective document image deblurring algorithm," in *Proc. CVPR*, 2011, pp. 369–376.
- [3] S. Cho, J. Wang, and S. Lee, "Handling outliers in non-blind image deconvolution," in *Proc. ICCV*, 2011, pp. 1–8.
- [4] R. Fergus, B. Singh, A. Hertzmann, S. Roweis, and W. Freeman, "Removing camera shake from a single photograph," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 787–794, Jul. 2006.
- [5] M. Ng, R. Plemmons, and S. Qiao, "Regularization of RIF blind image deconvolution," *IEEE Trans. Image Process.*, vol. 9, no. 6, pp. 1130–1134, Jun. 2000.
- [6] W. Richardson, "Bayesian-based iterative method of image restoration," *J. Opt. Soc. Amer.*, vol. 62, no. 1, pp. 55–59, 1972.
- [7] U. Schmidt, K. Schelten, and S. Roth, "Bayesian deblurring with integrated noise estimation," in *Proc. CVPR*, 2011, pp. 2625–2632.
- [8] Q. Shan, J. Jia, and A. Agarwala, "High-quality motion deblurring from a single image," *ACM Trans. Graph.*, vol. 27, no. 3, pp. 73–1–73–10, Aug. 2008.
- [9] L. Xu and J. Jia, "Two-phase kernel estimation for robust motion deblurring," in *Proc. 11th ECCV*, 2010, pp. 157–170.
- [10] H. Zhang, J. Yang, Y. Zhang, N. Nasrabadi, and T. Huang, "Close the loop: Joint blind image restoration and recognition with sparse representation prior," in *Proc. IEEE ICCV*, Nov. 2011, pp. 770–777.
- [11] D. Zoran and Y. Weiss, "From learning models of natural image patches to whole image restoration," in *Proc. IEEE ICCV*, Nov. 2011, pp. 479–486.
- [12] L. Yuan, J. Sun, L. Quan, and H. Shum, "Image deblurring with blurred/noisy image pairs," *ACM Trans. Graph.*, vol. 26, no. 3, pp. 1–8, Jul. 2007.
- [13] S. Zhuo, D. Guo, and T. Sim, "Robust flash deblurring," in *Proc. CVPR*, 2010, pp. 2440–2447.
- [14] J. Cai, H. Ji, C. Liu, and Z. Shen, "Blind motion deblurring using multiple images," *J. Comput. Phys.*, vol. 228, no. 14, pp. 5057–5071, Aug. 2009.
- [15] J. Cai, H. Ji, C. Liu, and Z. Shen, "High-quality curvelet-based motion deblurring from an image pair," in *Proc. CVPR*, 2009, pp. 1566–1573.
- [16] J. Chen, L. Yuan, C. Tang, and L. Quan, "Robust dual motion deblurring," in *Proc. CVPR*, 2008, pp. 1–8.
- [17] G. Harikumar and Y. Bresler, "Perfect blind restoration of images blurred by multiple filters: Theory and efficient algorithms," *IEEE Trans. Image Process.*, vol. 8, no. 2, pp. 202–219, Feb. 1999.
- [18] W. Li, J. Zhang, and Q. Dai, "Exploring aligned complementary image pair for blind motion deblurring," in *Proc. CVPR*, 2011, pp. 273–280.
- [19] A. Levin, Y. Weiss, F. Durand, and W. Freeman, "Understanding and evaluating blind deconvolution algorithms," in *Proc. CVPR*, 2009, pp. 1964–1971.
- [20] N. Joshi, S. Kang, C. Zitnick, and R. Szeliski, "Image deblurring using inertial measurement sensors," *ACM Trans. Graph.*, vol. 29, no. 4, pp. 1–8, Jul. 2010.
- [21] Q. Shan, W. Xiong, and J. Jia, "Rotational motion deblurring of a rigid object from a single image," in *Proc. 11th ICCV*, Oct. 2007, pp. 1–8.
- [22] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce, "Non-uniform deblurring for shaken images," in *Proc. CVPR*, Jun. 2010, pp. 491–498.
- [23] P. Favaro, M. Burger, and S. Soatto, "Scene and motion reconstruction from defocused and motion-blurred images via anisotropic diffusion," in *Proc. 8th ECCV*, 2004, pp. 257–269.
- [24] M. Sorel and J. Flusser, "Space-variant restoration of images degraded by camera motion blur," *IEEE Trans. Image Process.*, vol. 17, no. 2, pp. 105–116, Feb. 2008.
- [25] L. Xu and J. Jia, "Depth-aware motion deblurring," in *Proc. ICCP*, 2012, pp. 1–8.
- [26] Y. Tai, N. Kong, S. Lin, and S. Shin, "Coded exposure imaging for projective motion deblurring," in *Proc. CVPR*, 2010, pp. 2408–2415.
- [27] A. Gupta, N. Joshi, C. L. Zitnick, M. F. Cohen, and B. Curless, "Single image deblurring using motion density functions," in *Proc. 11th ECCV*, 2010, pp. 171–184.
- [28] S. Harmeling, M. Hirsch, and B. Schölkopf, "Space-variant single-image blind deconvolution for removing camera shake," in *Advances in Neural Information Processing System*. Cambridge, MA, USA: MIT Press, 2010.
- [29] M. Hirsch, C. Schuler, S. Harmeling, and B. Schölkopf, "Fast removal of non-uniform camera shake," in *Proc. ICCV*, 2011, pp. 463–470.
- [30] H. Lee, J. Kwon, and K. Lee, "Simultaneous localization, mapping and deblurring," in *Proc. ICCV*, 2011, pp. 1203–1210.
- [31] M. Ben-Ezra and S. K. Nayar, "Motion-based motion deblurring," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 6, pp. 689–698, Jun. 2004.
- [32] F. Li, J. Yu, and J. Chai, "A hybrid camera for motion deblurring and depth map super-resolution," in *Proc. CVPR*, 2008, pp. 1–8.
- [33] Y.-W. Tai, H. Du, M. S. Brown, and S. Lin, "Image/video deblurring using a hybrid camera," in *Proc. CVPR*, 2008, pp. 1–8.
- [34] G. Zhang, J. Jia, T. Wong, and H. Bao, "Consistent depth maps recovery from a video sequence," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 6, pp. 974–988, Jun. 2009.
- [35] D. Scharstein and R. Szeliski, "High-accuracy stereo depth maps using structured light," in *Proc. CVPR*, 2003, pp. 195–202.
- [36] C. Plagemann, V. Ganapathi, D. Koller, and S. Thrun, "Real-time identification and localization of body parts from depth images," in *Proc. ICRA*, 2010, pp. 3108–3113.
- [37] A. Letouze, B. Petit, E. Boyer, and M. Team, "Scene flow from depth and color images," in *Proc. BMVC*, 2011, pp. 46–1–46–11.
- [38] J. Shotton et al., "Real-time human pose recognition in parts from single depth images," in *Proc. CVPR*, 2011, pp. 1297–1304.
- [39] A. Kolb, E. Barth, and R. Koch, "ToF-sensors: New dimensions for realism and interactivity," in *Proc. CVPR*, 2008, pp. 1–6.
- [40] Microsoft Corporation, Albuquerque, NM, USA. (2014, May 3). *Kinect for Xbox 360* [Online]. Available: <http://www.xbox.com/en-GB/Kinect>
- [41] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, U.K.: Cambridge Univ. Press, 2003.
- [42] S. Cho and S. Lee, "Fast motion deblurring," *ACM Trans. Graph.*, vol. 28, no. 5, p. 145, 2009.
- [43] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *Int. J. Comput. Vis.*, vol. 47, no. 1, pp. 7–42, 2002.
- [44] A. Levin, R. Fergus, F. Durand, and W. Freeman, "Image and depth from a conventional camera with a coded aperture," *ACM Trans. Graph.*, vol. 26, no. 3, pp. 1–10, Jul. 2007.
- [45] I. Daubechies, R. DeVore, M. Fornasier, and C. Güntürk, "Iteratively reweighted least squares minimization for sparse recovery," *Commun. Pure Appl. Math.*, vol. 63, no. 1, pp. 1–38, Jan. 2010.
- [46] C. Stewart, "Robust parameter estimation in computer vision," *Soc. Ind. Appl. Math. Rev.*, vol. 41, no. 3, pp. 513–537, Sep. 1999.
- [47] R. Köhler, M. Hirsch, B. Mohler, B. Schölkopf, and S. Harmeling, "Recording and playback of camera shake: Benchmarking blind deconvolution with a real-world database," in *Proc. 12th ECCV*, 2012, pp. 27–40.
- [48] D. Min and K. Sohn, "Cost aggregation and occlusion handling with WLS in stereo matching," *IEEE Trans. Image Process.*, vol. 17, no. 8, pp. 1431–1442, Aug. 2008.

Tao Yue received the B.S. degree in automation from Northwestern Polytechnical University, Xi'an, China, in 2009. He is currently pursuing the Ph.D. degree with the Department of Automation, Tsinghua University, Beijing, China. His research interests mainly include image processing and computational photography.





Jinli Suo received the B.S. degree in computer science from Shandong University, Shandong, China, in 2004, and the Ph.D. degree from the Graduate University of Chinese Academy of Sciences, Beijing, China, in 2010. She is currently a Lecturer with the Department of Automation, Tsinghua University, Beijing. Her research interests mainly include computer vision, computational photography, and statistical learning.



Qionghai Dai (SM'05) received the Ph.D. degree in automation from Northeastern University, Shenyang, China, in 1996. He has been with the Department of Automation, Tsinghua University, Beijing, China, as a Faculty Member since 1997 and a Professor since 2005. He has authored more than 120 conference and journal papers. He holds 67 patents. His current research interests include computational photography, computational optical sensing, and compressed sensing imaging and vision. His work is motivated by challenging applications in the fields of computer vision, computer graphics, and robotics.