Efficient 3D Kernel Estimation for Non-uniform Camera Shake Removal Using Perpendicular Camera System

Tao Yue
yue-t09@mails.tsinghua.edu.cn

Jinli Suo Qionghai Dai
{jlsuo, qhdai}@tsinghua.edu.cn

Department of Automation, Tsinghua University, Beijing

Abstract

Non-uniform camera shake removal is a knotty problem which plagues the researchers due to the huge computational cost of high-dimensional blur kernel estimation. To address this problem, we propose an acceleration method to compute the 3D projection of 2D local blur kernels fast, and then derive the 3D kernel by interpolating from a minimal set of local blur kernels. Under this scheme, a perpendicular acquisition system is proposed to increase the projection variance for reducing the ill-posedness of 3D kernel estimation. Finally, based on the minimal 3D kernel solver, a RANSAC-based framework is developed to raise the robustness to estimation error of 2D local blur kernels. In experiments, we validate the effectiveness and efficiency of our approach on both synthetic and real captured data, and promising results are obtained.

1. Introduction

Image blur caused by camera shake is a common degradation in photography, and many deblurring methods have been proposed in the past decades. Conventional deblurring methods[2][3][5][15][22][23] assume a uniform blur kernel and formulate the blur process as 2D convolution. However, this assumption rarely holds in real cases[19], and non-uniform blur caused by camera shake is drawing increasing attentions. Several models and deblurring algorithms are proposed to formulate the non-uniformity and to solve the problem as fast and robust as possible.

Some researchers[18][4] adopt a piecewise-uniform blur model to reduce the computation burden of the problem. These works are particularly suitable for dealing with blur caused by object motion, since in such cases the same object (part) shares similar motion patterns and thus similar LBKs, i.e., locally invariant. However, these methods do not work well for the non-uniform blur caused by camera shake, because blur kernels may vary continuously and thus cannot be described by the piecewise-uniform model well enough.

Some other researchers attempt to use a unified camera motion to constrain the spatially varying 2D blur kernels. In these works, the blur kernels are unified in a subspace of the 6D motion parameter space. Shan *et al*. [24] assume the 1D in-plane rotation and thus can only deal with the image blur caused by camera rotation around z-axis. To remove more general non-uniform blur, 3D motion blur models are proposed. Whyte *et al*. [28] take all the 3 rotations into consideration and Gupta *et al*. [7] use in-plane rotation and x, y-translation as the Degree of Freedom (DoF) of the camera motion, but these approaches suffer from the intensive computation due to its the high dimensionality of parameter space. To raise the efficiency, Hirsch *et al*. [9] propose efficient filter flow, while Hu and Yang propose a back projection and intersection based method for fast kernel estimation/initialization. However, both methods are not robust enough and still needs to be incorporated in a time consuming iterative optimization framework. Differently, our estimation eliminates iterative optimization due to superior efficiency and robustness, and hence shortens the running time significantly.

With the development of computational photography, non-uniform deblurring benefits from the aid of non-traditional imaging mechanisms. For example, Joshi *et al*. [14] use inertial measurement sensors to capture the 6D camera motion directly to assist deblurring, Tai *et al*. [26] propose coded exposure imaging system to solve 6D motion blur, Ezra *et al*. [1], Tai *et al*. [25] and Li *et al*. [20] use low-rate high-resolution / high-rate low-resolution hybrid camera systems to facilitate blur kernel estimation. There are some limitations affecting the final performance in above works, such as the drift and noise of observed signal in [14], indispensable user interactions in [26], and potential inaccurate optical flow in hybrid systems[1][25][20]. In spite of the limitations in these works, introducing additional information by computational photography systems is a worth considering approach for camera shake removal.

This paper addresses the high computational cost of 3D blur kernel estimation by intersecting the 3D projections of

a series of 2D Local Blur Kernels (LBKs) estimated from image patches. By linearizing camera motion model around the center of each local blur model, we propose a warping based-method to compute the 3D projection efficiently. However, the performance of intersection may suffer from ill-posedness caused by limited variance among LBKs (especially under a small field of view) and estimation error of LBKs, these problems inspire us to propose two strategies incorporated in a framework illustrated in Fig. 1.

Firstly, after analyzing the relation between angle of view and ill-posedness, we design a binocular perpendicular acquisition system to enlarge the angle. Then the 3D camera motion is derived under a RANSAC based iteration of two processes—projecting LBKs into 3D motion parameter space and performing computational efficient intersection. The RANSAC framework here raises the robustness to estimation errors of LBKs. Finally, a non-blind deblurring algorithm is applied to restore the latent sharp image.

In summary, this paper contributes in following aspects: (i) We propose an efficient method to accelerate the 3D projecting from LBKs. (ii) The minimal solution of 3D kernel estimation from two LBKs by intersecting are proposed and a RANSAC-based framework is introduced to increase the robustness. (ii) A perpendicular camera system is proposed to capture approximately perpendicular 2D projections of 3D camera motion to reduce the ill-posedness.

2. Efficient 3D projection of LBKs

We adopt the 3D rotation blur model proposed by Whyte *et al*. [28], which is approximately depth independent. According to perspective geometry, the 2D blur kernels would vary continuously in such cases, so we can approximate the 2D blur kernels within a patch by a uniform pattern, we name which Local Blur Kernels (LBKs).

LBKs can be regarded as the 2D projections of a high-dimensional camera motion, according to perspective geometry[8] we define the mapping from a 3D rotation kernel to its 2D LBK within a certain patch by

$$K_2^{\mathbf{x}}(\boldsymbol{m}) = \sum_{\theta} \mathbf{1}(\boldsymbol{m}'(\boldsymbol{\theta}, \mathbf{x}), \boldsymbol{m}) K_3(\boldsymbol{\theta}), \tag{1}$$

where $K_2^{\mathbf{x}}(\boldsymbol{m})$ is the 2D blur kernel at coordinate $\mathbf{x}$, and $\boldsymbol{m}$ indexes the position in $K_2^{\mathbf{x}}$; $K_3(\boldsymbol{\theta})$ denotes the 3D camera motion, with $\boldsymbol{\theta} = (\theta_x, \theta_y, \theta_z)$ being the rotation around x, y, z-axis; $\mathbf{1}(\cdot)$ is the indicator to label whether the mapping $\boldsymbol{m}'$ of pixel $\mathbf{x}$ with camera pose $\boldsymbol{\theta}$ is equal to $\boldsymbol{m}$. Here $\boldsymbol{m}'(\cdot)$ is the mapping function (see [8, 28] for details).

To compute projection of K_2 in 3D parameter space fast, we extract a slice of K_3 with constant θ_z and assume the support region of 3D kernel is located on this slice, then

$$K_3(\boldsymbol{\theta})|_{\theta_z} = K_2^{\mathbf{x}}(\boldsymbol{m}'(\boldsymbol{\theta})|_{\theta_z}). \tag{2}$$

Linearizing $\boldsymbol{m}'(\cdot)$ by Taylor's expansion, Eq. 2 becomes

$$K_3(\boldsymbol{\theta})|_{\theta_z} = K_2^{\mathbf{x}}(\boldsymbol{m}'_0|_{\theta_z} + \frac{\partial \boldsymbol{m}'}{\partial \theta_x} \cdot \theta_x + \frac{\partial \boldsymbol{m}'}{\partial \theta_y} \cdot \theta_y). \tag{3}$$

Here, $\frac{\partial \boldsymbol{m}'}{\partial \theta_x}$ and $\frac{\partial \boldsymbol{m}'}{\partial \theta_y}$ are the partial derivatives of $\boldsymbol{m}'(\cdot)$ with respect to θ_x and θ_y, and $\boldsymbol{m}'_0|_{\theta_z}$ is the constant value with $\boldsymbol{\theta} = (0, 0, \theta_z)$. When $\theta_z = 0$, $\boldsymbol{m}'_0|_{\theta_z}$ equals to 0 and the slice is a parallalogram deformation of K_2 and can be computed fast by resampling K_2. When θ_z doesn't equal 0, the slice can be viewed as a translation version of slice $K_3|_{\theta_z=0}$. Therefore, we can compute slice $K_3|_{\theta_z=0}$ first by resampling K_2 and translate the slice with vector $\boldsymbol{m}'_0|\theta_z$ to derive any slice of K_3. Stacking all the slices in order, we can compute the 3D projection of 2D LBK very fast, with a much higher efficiency than [11].

Given several LBKs, we can compute two 3D projection of the LBKs, and then the 3D camera motion can be derived by intersecting these projections in parameter space under the assumption of 1D manifold motion trace. Fig. 1 (2.2) visualizes this procedure: If the motion trace lie on a 1D manifold, the 3D projection of non-zero entries in a LBK compose a surface in 3D space and the latent motion trajectory slides across the surface from one side to another. Naturally, the 3D camera motion is exactly the intersection of these surfaces projected from different LBKs.

However, it is non-trivial to compute the 3D motion accurately by intersecting these projections directly for mainly two reasons: (1) Limited by the angle of view, the angle between LBKs from a single image are quite small, which means the two surfaces in 3D space intersact with a small angle. It is well known that intersection with small angle will introduce large ill-posedness. (2) Because of the low quality or textureless-ness of some image patches, even state-of-the-art uniform deblurring algorithms cannot avoid incorrect or inaccurate LBKs, so that the result cannot avoid suffering from outliers.

To address above two problems, this paper attempts to obtain sufficiently dissimilar LBKs and keeps the problem well-posed by the aid of a perpendicular camera system. Accordingly, we propose the 2-LBK intersection based 3D kernel solver, which will be discussed in details in Sec. 3. Then a RANSAC based optimization framework is proposed to get rid of LBK estimation errors in Sec. 4.

3. 3D Kernel Estimation by Perpendicular Intersection

3.1. Why Perpendicular Intersection?

Intuitively, multiple cameras (fastened together) tend to provide more diverse projections compared to one single camera and thus raise the robustness of blur kernel estimation. This inspires us to build a camera-pair system for fast

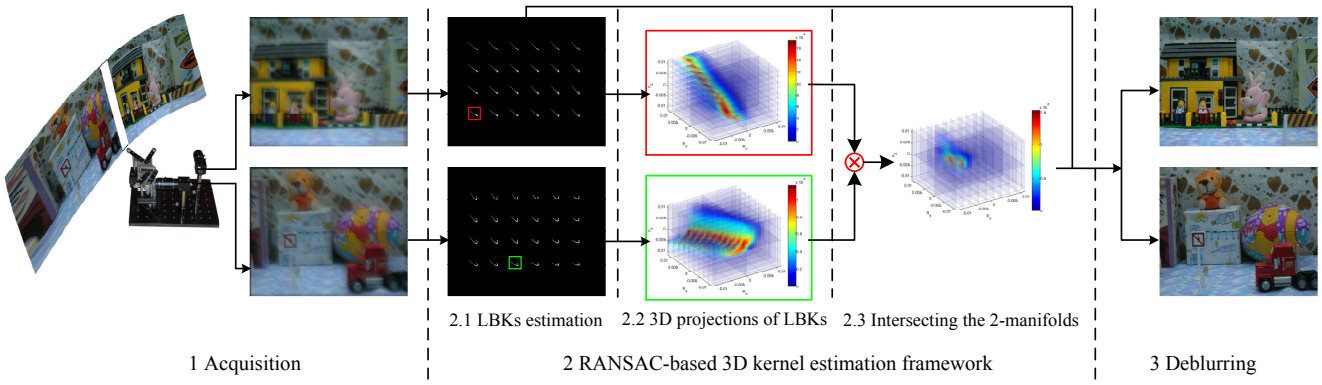


Figure 1. **Diagram of our system.**

camera shake removal. For a quantitative analysis, we compute the condition numbers of the linear equation systems built on camera pairs (composed by Eq. 1) with different between-camera angles, as illustrated in Fig. 2. We can see that the condition number is extremely huge for small between-camera angle, and reduces rapidly as the angle increases. In other words, Eq. 1 is better posed when the between-manifold angle is larger. Unfortunately, due to the limited angle of view and errors in LBK estimation, it is often hard to find two well estimated LBKs within a single image and with a large intersection angle between their 3D projections. To keep the problem well-posed, we propose a perpendicular camera system which tries to make the intersection angle approximately perpendicular and thus a corresponding 3D kernel solver.

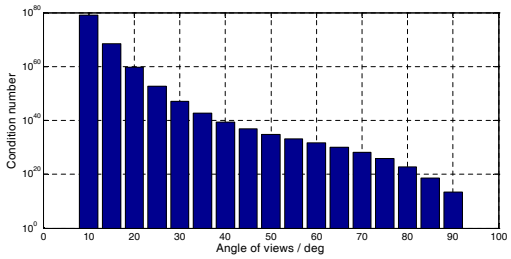


Figure 2. **Condition number of linear system defined in Eq. 1 at different between-view angles.**

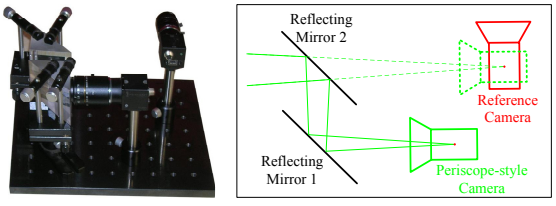
Note that, although the perpendicular layout is recommended, the approach can also be applied to single view image or non-perpendicular biocular system, and gives reasonable results, as shown in experiments in Sec. 5.

3.2. Perpendicular Acquisition System

Fig. 3 shows the prototype of our perpendicular camera system and its light path. The system consists of two cameras (Point Grey FL2-08S2C) and two reflecting mirrors. As shown in Fig. 3(b), the green camera can be transformed to the position of the dashed green camera, so the optical centers of two cameras can be coincident exactly. To facilitate the description, the red camera is called reference camera and the green one is called periscope-style camera.

To ensure that the two cameras are perpendicular and the optical centers are coincident, we firstly turn the reference camera 90° counterclockwise (can be controlled by rotation base), so that the cameras are parallel with each other. Secondly, we fix the reference camera and capture a snapshot without the reflecting mirrors, then we turn back the reference camera and tune the periscope-style camera and two reflecting mirrors to make sure that the periscope-style camera capture the same view with the snapshot taken by the reference camera. During the adjustment, the optical base of the system should be immovable.

It is worth noting that it is difficult to align two optical centers exactly by above calibration, here we analyze quantitatively the affects from inaccurate alignment. Assuming the camera offset causing misalignment lying on the rotating axis of the camera bases and the eccentric offset being Δr, a rotation $\Delta\theta$ will cause a translation $\Delta r\Delta\theta$. Consequently, denoting focal length as f and scene depth as D, the eccentric offset will result in movement $\Delta r\Delta\theta f/D$ (in pixels) in CCD plane. Empirically, $\Delta\theta f$ includes dozens of pixels (e.g., 30), we just need $\Delta r < D/30$ to ensure that the offset will not cause a movement larger than a pixel in CCD plane. Considering the camera size, it is reasonable to assume an offset no larger than 1cm, and it can be ignored for scenes farther than 30cm.



(a) (b)

Figure 3. **Our perpendicular camera system and its light path.**

3.3. Perpendicular Intersection

Based on above acquisition system, we develop the corresponding axes-exchanging image blur model and propose a 2-LBK intersection based estimator of 3D blur kernel.

Axes-exchanging blur model. The axes-exchanging blur model exchanges the axes in turn to unify the parameters of two perpendicular views into one space.

Under rotation motion assumption, the view taken by a camera during a short exposure can be regarded as a homography transformation of the latent image. Since camera intrinsics are supposed known, the homography H only depends on the 3×3 rotation matrix $\mathbf{R}$. For our perpendicular camera system, the homography pair of corresponding views can be respectively computed by

$$\begin{aligned} H &= \mathbf{A}\mathbf{R}\mathbf{A}^{-1} \\ H' &= \mathbf{A}'\mathbf{R}'\mathbf{A}'^{-1}. \end{aligned} \tag{4}$$

Here $\mathbf{A}$ and $\mathbf{A}'$ are intrinsics of two perpendicular cameras.

As mentioned in Sec. 2, rotation matrix $\mathbf{R}$ can be represented by matrix exponential $\mathbf{R} = e^{[\boldsymbol{\theta}]_\times}$, where $\boldsymbol{\theta}$ is the angle-axis rotation vector $(\theta_x, \theta_y, \theta_z)^{\mathrm{T}}$. For the counterpart matrix $\mathbf{R}'$, the angle-axis rotation vector $\boldsymbol{\theta}'$ can be derived by exchanging $x-$ and z-axis and inverting the sign of the last element, i.e. $\boldsymbol{\theta}' = (\theta_z, \theta_y, -\theta_x)$.

Then, for two image patches sampled from a blurry image pair with center coordinates being $\mathbf{x}$ and $\mathbf{x}'$ respectively, the mapping from 3D kernel to the 2D LBKs are:

$$\begin{cases} K_2^{\mathbf{x}}(\boldsymbol{m}) = \sum_\theta \mathbf{1}(H(\boldsymbol{\theta})\mathbf{x}, \boldsymbol{m}) K_3(\boldsymbol{\theta}) \\ K_2'^{\mathbf{x}'}(\boldsymbol{n}) = \sum_\theta \mathbf{1}(H'(\boldsymbol{\theta})\mathbf{x}', \boldsymbol{n}) K_3(\boldsymbol{\theta}) \end{cases}. \tag{5}$$

2-LBK intersection based 3D kernel estimation. Generally, above equation system is severely ill-posed, so we need to introduce reasonable priors as additional constraints. The sparsity of blur kernel has already been used in previous works [5][2][17][29], but most of which introduce the prior into an objective function as the regularization term, which alleviates the ill-posedness, but significantly increases the computational complexity of interactive optimization.

Instead, our method assumes the 3D kernel to be a high-dimensional curve which can be derived by intersecting 3D projections of LBKs. As a matter of fact, the widely used sparsity prior of the blur kernel is implied here. After determining the 3D trace of the camera motion, we further set a value to each 3D kernel element to represent the time elapse during exposure, which is exactly the same as the intensity of its projection in LBKs. In addition, due to the inaccuracy of LBKs estimation and the discretization error, we use the average intensity of the projection points on the LBK pair as the estimation value of the 3D kernel point.

We first compute 3D projections of LBK pair at positions randomly selected from perpendicular blurred image pair by our fast 3D projecting method. Then, the support region of the 3D kernel can be computed by thresholded binarization to the product of 3D projections of the LBK pair as

$$SK_\sigma = \{\boldsymbol{\theta} : P_{K_2^{\mathbf{x}}}(\boldsymbol{\theta}) P_{K_2'^{\mathbf{x}'}}(\boldsymbol{\theta}) \geq \sigma\} \tag{6}$$

where σ is the threshold determined by the minimum element in 3D kernel, and is set to be 0.001 here. Finally, each non-zero element of the 3D kernel are estimated by

$$K_3(\boldsymbol{\theta}) = (P_{K_2^{\mathbf{x}}}(\boldsymbol{\theta}) + P_{K_2'^{\mathbf{x}'}}(\boldsymbol{\theta}))/2 \quad \forall \boldsymbol{\theta} \in SK_\sigma. \tag{7}$$

Note that the intersection approach implies that there does not exist any self-intersection in a LBK. In spite that this does not hold sometimes, the number of self-intersection points is extremely small compared to the size of blur kernel, so it does not affect the performance. The similar property is also utilized for motion segmentation from motion blurring[21].

4. RANSAC Based Estimation Framework

Because the LBKs estimated from textureless regions may be inaccurate or even wrong, we apply a RANSAC based framework to raise the robustness of 3D kernel estimation. Furthermore, to adjust our framework towards non-uniform deblurring, we modify the traditional RANSAC framework by introducing a novel assessment function for candidate solutions. Generally, RANSAC evaluates the quality of candidate solutions by the size of inlier set, which is unsuitable for deblurring since the difference among image regions are neglected. Considering that different patches have varying saliency and their corresponding LBKs have different confidences, we propose an assessment function considering both factors.

Confidence. Because the LBKs estimated from local patches suffer from errors and inaccuracy, we define a measurement to assign a confidence weight to each LBK in the assessment function. As discussed by Gupta *et al*. [7], the most important factor affecting the quality of deblurred results is the texture richness of the patches, so we use the same metric as Gupta *et al*. do (i.e. the average of Harris corner metric of all the pixels in the patch).

Saliency. Since our ultimate goal is to restore the latent sharp image instead of deriving the real 3D kernel, we prefer a 3D kernel working better in the important regions of an image. It is known that human eyes are sensitive to edges with high contrast or salient regions distinctive from its surroundings, and there are plenty of models for computing image saliency map, e.g., [6], [10], [13] and [27]. Considering the computational cost, we select Guo *et al*.'s [6] method which computes the saliency fast in frequency domain.

Combing above two factors, our assessment function for each patch turns into:

$$\begin{aligned} f(K_3) &= \sum_{i \in \Omega} w(x) d(K_2^x, P(K_3, x)) \\ \Omega &= \{x | d(K_2^x, P(K_3)) > thr\} \\ w(x) &= C(x)S(x). \end{aligned} \tag{8}$$

In the above equations, K_3 is the 3D blur kernel and $P(\cdot)$ is the function defined by Eq. 1 which maps a 3D kernel to its 2D projections; K_2^x is the 2D blur kernel estimated at pixel x; $d(K_2^x, P(K_3))$ is the distance between 2D projection of K_3 and the estimated 2D LBK K_2^x at position x. To increase robustness, LBK at pixel x is given a weight $w(x)$ to penalize bad LBK estimation, with $S(x)$ and $C(x)$ respectively denoting saliency and confidence of patch x; Ω is the inlier set in RANSAC strategy and thr is the threshold for inlier selection; $d(\cdot)$ is defined by normalized cross-correlation as Hu *et al*. [12]'s,

$$d(K_2^i, P(K_3)) = max_{\boldsymbol{\lambda}} \frac{K_2^i(\mathbf{m}) \cdot P(K_3, \mathbf{m} + \boldsymbol{\lambda})}{||K_2^i|| \cdot ||P(K_3)||}. \quad (9)$$

Here, $||\cdot||$ is l_2 norm, $\boldsymbol{\lambda}$ denotes a slight shift used to eliminate the translation ambiguity of 2D LBKs.

Different from the framework used by Gupta [7], our method prefers to process the blur kernel rather than the deblurred patches. The 3D kernel is directly computed from the inlier set of LBKs, so we do not do the EM-style iteration which optimizes the latent image L and 3D kernel K_3 iteratively. In addition, since the perpendicular camera system is used to capture image pairs, we can solve the 3D kernel with a minimum number of LBKs, i.e. we adopt the minimal solution (2 patches, one from each camera) as basic solver. Therefore, our method can be more error-tolerant, since selecting a small data set without outliers is much easier. Profiting from the above properties, our RANSAC based framework could be efficient and effective for 3D kernel estimation.

In our experience, Cho and Lee [3]'s program can give pretty good results from 120$\tilde{2}$00 pixels wide square patches with proper textures in less than 0.5s. We select the patches uniformly and make sure each patch is overlapped with its neighbors. Therefore, about less than 200 patches are needed for a 1024×768 image, and about 100s for computing the LBKs from this patch.

5. Experimental Results and Analysis

We conduct a series of experiments to validate the proposed fast shake removal approach on both synthetic and real captured data. Theoretically any uniform deblurring can be applied in LBKs estimation and any non-uniform deblurring can be used in final restoration, considering the computation complexity, we use Cho and Lee [3]'s method to estimate the LBKs and Hirsch *et al*.'s[9] efficient filter flow based method (a non-uniform variant of Krishnan and Fergus [16]'s algorithm) to do the final deblurring.

Experiment on Synthetic Data. We first test the performance of our approach on synthetic data. To verify the effectiveness of our kernel estimator, we test the algorithm on the synthetic LBKs of two perpendicular views and with large spatially-variance in each single view. The 3D-rotation camera motion are generated randomly and with large spatially-variation, and Fig. 4 shows the ground truth blur kernel maps and our estimated results respectively. The results shows that our perpendicular intersection based kernel solver can give exact results for the error free data.

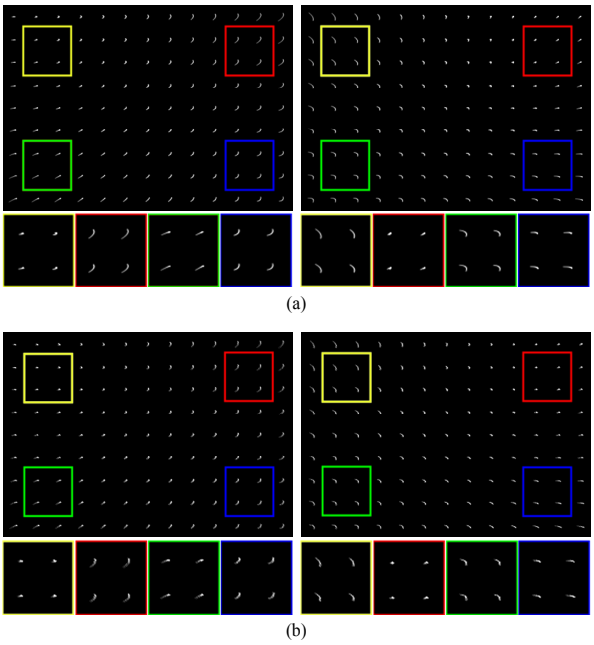


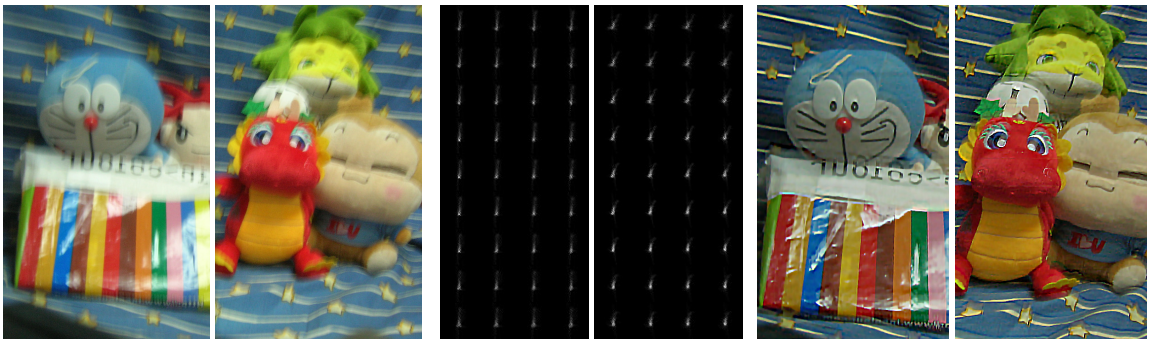
Figure 4. **Kernel estimation on synthetic data.** Top row: two views' ground truth kernels. Bottom row: estimated kernels from each single view.

Experiment on Real Captured Data. Here we give an example of shake removal on a real captured image pair by our prototype to verify the effectiveness and robustness of the whole system. From the captured blurry image pair in Fig. 5(a) we can see that there exist apparent difference between the blur effects in two views, which provide two 2D projections of the latent 3D blur kernel along two different directions, and help intersection algorithm. The estimated blur kernels of two views are shown in Fig. 5(b) and also validate the difference between two 2D projections of the 3D blur kernel. The final deblurring result of our proposed method is shown in Fig. 5(c), which shows that our algorithm achieves good performance in both views while at low computational cost.

6. Summary and Future Work

We present an intersection based approach and a perpendicular acquisition system for estimating 3D blur kernels efficiently and effectively. These advantages are attributed to largely different 2D projection directions of the 3D kernel and the RANSAC framework.

The algorithm is limited in two aspects: (i) cannot deal



(a) (b) (c)

Figure 5. **Deblurring result on real captured data.** (a) Blurry image pair. (b) Blur kernels estimated by perpendicular intersection. (c) Final deblurred results.

with the translation caused image blur which is depth dependent; (ii) to a certain extent, the performance of our algorithm depends on LBK estimation. Therefore, we will try to take depth information into consideration and further raise the accuracy of LBKs estimation. In the future, we will try to extend our system for handling all the 6 DoFs of camera motion. It is worth noting that extension to a higher DoF is feasible due to high efficiency.

Acknowledgement

This work was supported by the Project of NSFC (No. 61327902 and 61171119).

References

[1] M. Ben-Ezra and S. Nayar. Motion-based motion deblurring. *IEEE Trans. PAMI*, 26(6):689–698, Jun. 2004.

[2] J. Cai, H. Ji, C. Liu, and Z. Shen. Blind motion deblurring from a single image using sparse approximation. In *CVPR*, 2009.

[3] S. Cho and S. Lee. Fast motion deblurring. *ACM Trans. Graphics (SIGGRAPH Asia 2009)*, 28(5), 2009.

[4] S. Cho, Y. Matsushita, and S. Lee. Removing non-uniform motion blur from images. In *ICCV*, 2007.

[5] R. Fergus, B. Singh, A. Hertzmann, S. Roweis, and W. Freeman. Removing camera shake from a single photograph. *ACM Trans. Graphics*, 25(3), Jul. 2006.

[6] C. Guo, Q. Ma, and L. Zhang. Spatio-temporal saliency detection using phase spectrum of quaternion fourier transform. In *CVPR*, 2008.

[7] A. Gupta, N. Joshi, C. L. Zitnick, M. F. Cohen, and B. Curless. Single image deblurring using motion density functions. In *ECCV*, 2010.

[8] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2003.

[9] M. Hirsch, C. Schuler, S. Harmeling, and B. Schölkopf. Fast removal of non-uniform camera shake. In *ICCV*, 2011.

[10] X. Hou and L. Zhang. Saliency detection: A spectral residual approach. In *CVPR*, 2007.

[11] Z. Hu and M.-H. Yang. Fast non-uniform deblurring using constrained camera pose subspace. In *ECCV*, 2012.

[12] Z. Hu and M.-H. Yang. Good regions to deblur. In *ECCV*, 2012.

[13] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. PAMI*, 20(11):1254–1259, 1998.

[14] N. Joshi, S. Kang, C. Zitnick, and R. Szeliski. Image deblurring using inertial measurement sensors. *ACM Trans. Graphics*, 29(4), Jul. 2010.

[15] N. Joshi, R. Szeliski, and D. Kriegman. Psf estimation using sharp edge prediction. In *CVPR*, june 2008.

[16] D. Krishnan and R. Fergus. Fast image deconvolution using hyper-laplacian priors. *Advances in Neural Information Processing Systems*, 22:1–9, 2009.

[17] D. Krishnan, T. Tay, and R. Fergus. Blind deconvolution using a normalized sparsity measure. In *CVPR*, 2011.

[18] A. Levin. Blind motion deblurring using image statistics. In *NIPS*, 2007.

[19] A. Levin, Y. Weiss, F. Durand, and W. Freeman. Understanding and evaluating blind deconvolution algorithms. In *CVPR*, 2009.

[20] F. Li, J. Yu, and J. Chai. A hybrid camera for motion deblurring and depth map super-resolution. In *CVPR*, 2008.

[21] C. Paramanand and A. Rajagopalan. Motion blur for motion segmentation. In *ICIP*, pages 4244–4248, Sept 2013.

[22] W. Richardson. Bayesian-based iterative method of image restoration. *Journal of the Optical Society of America*, 62(1):55–59, 1972.

[23] Q. Shan, J. Jia, and A. Agarwala. High-quality motion deblurring from a single image. *ACM Trans. Graphics*, 27(3), Aug. 2008.

[24] Q. Shan, W. Xiong, and J. Jia. Rotational Motion Deblurring of a Rigid Object from a Single Image. In *ICCV*, 2007.

[25] Y. Tai, H. Du, M. Brown, and S. Lin. Image/video deblurring using a hybrid camera. In *CVPR*, 2008.

[26] Y. Tai, N. Kong, S. Lin, and S. Shin. Coded exposure imaging for projective motion deblurring. In *CVPR*, 2010.

[27] D. Walther and C. Koch. Modeling attention to salient proto-objects. *Neural Networks*, 19(9):1395–1407, 2006.

[28] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce. Non-uniform deblurring for shaken images. In *CVPR*, 2010.

[29] 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 *ICCV*, 2011.