Image Deblurring using Smartphone Inertial Sensors

Zhe Hu[1], Lu Yuan[2], Stephen Lin[2], Ming-Hsuan Yang[1]
[1]University of California, Merced [2]Microsoft Research
https://eng.ucmerced.edu/people/zhu/CVPR16_sensordeblur.html/

Abstract

Removing image blur caused by camera shake is an ill-posed problem, as both the latent image and the point spread function (PSF) are unknown. A recent approach to address this problem is to record camera motion through inertial sensors, i.e., gyroscopes and accelerometers, and then reconstruct spatially-variant PSFs from these readings. While this approach has been effective for high-quality inertial sensors, it has been infeasible for the inertial sensors in smartphones, which are of relatively low quality and present a number of challenging issues, including varying sensor parameters, high sensor noise, and calibration error. In this paper, we identify the issues that plague smartphone inertial sensors and propose a solution that successfully utilizes the sensor readings for image deblurring. With both the sensor data and the image itself, the proposed method is able to accurately estimate the sensor parameters online and also the spatially-variant PSFs for enhanced deblurring performance. The effectiveness of this technique is demonstrated in experiments on a popular mobile phone. With this approach, the quality of image deblurring can be appreciably raised on the most common of imaging devices.

1. Introduction

A tremendous number of photographs are captured each day using smartphones. Unfortunately, camera shake on these light, hand-held devices is a common occurrence that results in blurry images. To remove the motion blur, both its point spread function (PSF), i.e., blur kernel, and the latent image need to be estimated from the blurred photograph, which is an ill-posed problem. Complicating this problem further is the fact that motion blur often varies spatially across the image in practice. Although there exist many algorithms for removing spatially-invariant motion blur, the spatially-variant deblurring problem remains unsolved, and it is difficult to remove such blur without additional information about the camera motion.

Unlike priors commonly used for latent images (e.g., those derived from natural image statistics), the distribution of camera motions is difficult to model due to its large diversity and inherent randomness. To gain additional information on such motion, inertial sensors such as gyroscopes and accelerometers have recently been exploited [15, 24, 19], with blur kernels estimated directly from the recorded camera motions and then used together with non-blind deconvolution techniques to recover the sharp latent image. These previous techniques assume the inertial sensor readings to be reliable enough for accurate blur kernel recovery. However, the inertial sensors in smartphones are far from ideal, as they are designed for tasks such as gaming that do not require high-level accuracy. Among the issues common to smartphone inertial sensors are a time delay between sensor and image measurements, and significant sensor noise that cannot be fully addressed through offline calibration. In addition, the rotational center of camera motion can vary with each photograph, and inaccurate accelerometer readings can accumulate into substantial errors in measured camera translation. These practical issues lead to a considerable challenge in using inertial sensors for deblurring smartphone photos.

In this paper, we present a solution that both manages these individual problems and compensates for the shortcomings in sensor data quality. Techniques are proposed for dynamically recovering the time delay between the inertial and camera sensors, calibrating the rotational center of camera motion online, and refining the blur kernels initially reconstructed from noisy sensor readings. Equally important, our method utilizes inertial sensor readings only as *guidance* for kernel estimation, rather than directly computing camera motion from the sensors as done in previous inertial sensor-based deblurring systems [15, 24, 19]. Our work jointly leverages the inertial sensor data and the captured image itself to overcome the limitations each may have in solving for spatially-variant PSFs. The experiments show that, even with smartphone inertial sensors, this approach can produce high-quality deblurring results, as exemplified for an Apple iPhone 6 in Figure 1.

This work presents a significant step towards the practical use of image deblurring. Existing deblurring algorithms

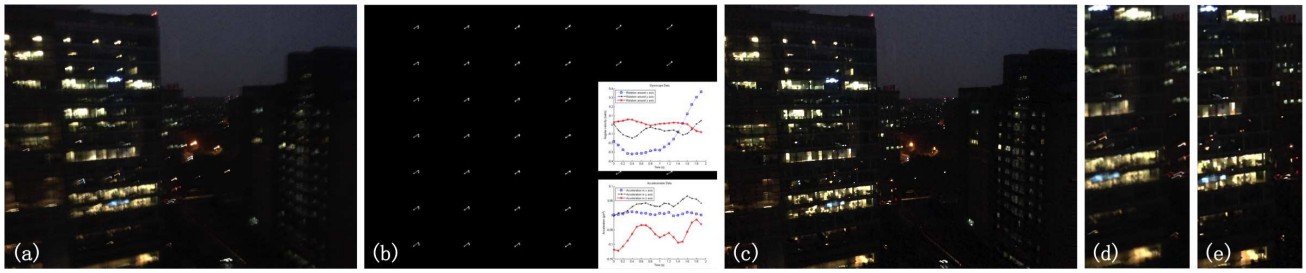


Figure 1. Example of smartphone image deblurring. (a) Blurry input image. (b) Estimated PSFs with the help of inertial sensors. (c) Recovered latent image. (d) A patch from the blurry image in (a). (e) The corresponding patch from the recovered latent image in (c).

have exhibited limited performance on real-world images captured with smartphones, which are by far the dominant device for digital photography. With effective use of smartphone inertial sensors, image deblurring can become a more practical and mainstream operation for millions of users.

2. Related Work

Significant progress in blind deconvolution has been witnessed in recent years (e.g., [20, 32, 25, 9, 30]). Since blind deconvolution is an ill-posed problem, prior knowledge or additional captured information is often required to obtain effective solutions.

The most common prior knowledge comes from natural image statistics, such as a heavy-tailed image gradient distribution [6, 22], alpha matte of the blurry edge [14], sparsity of wavelet coefficients [2], and adaptive texture priors [5]. Besides priors on image statistics, knowledge in the form of camera motion models has been used in solving more complicated spatially-variant blur. The blurry image can be represented as the integration of all the intermediate images captured by the camera along the motion trajectory [29, 26]. The camera motion can be modeled by three dimensions of camera rotation [29] or 6D camera motion with an in-plane translation and rotation [8]. To speed up the optimization step, fast patch-based non-uniform deblurring methods have been developed [10, 11]. These forms of prior knowledge have been shown to be helpful, but they often do not adequately constrain the deblurring result by themselves.

Another line of research tackles image deblurring by leveraging auxiliary information acquired at capture time [21, 31, 17, 3, 1, 27], for example, blurred and noisy image pairs [31], or high-resolution blurred and low-resolution sharp image pairs [1, 27].

More accurate information about camera motion can be obtained using inertial sensors such as gyroscopes and accelerometers, which have been integrated into many ordinary mobile devices. Šindelář and Šroubek [24] proposed a deblurring system for smartphones that synthesizes local blur kernels based on gyroscope readings. The system compares favorably to recent deblurring algorithms, but requires user interaction to synchronize the camera and gyroscope sensors. Joshi et al. [15] built an elaborate DSLR-based camera system with gyroscope and accelerometer. The inertial sensors are assumed to have been calibrated and synchronized beforehand. A similar approach for gyroscope calibration was employed by Park and Levoy [19] for the multiple image deblurring problem. Since noisy sensor readings cause drifting problems, both methods [15, 19] correct the sensor parameters using image priors or multiple images. All of these works assume their recorded sensor data to be reliable, or only slightly relax this assumption. However, there are more challenging issues in using smartphone inertial sensors, which cannot provide high-quality sensor data for effective image deblurring. By contrast, our method does not directly estimate camera motion from the sensors as done in these prior techniques [24, 15]. We only consider sensor readings as *guidance* for kernel estimation, and use image priors for further kernel refinement. Instead of offline calibration commonly used before, we utilize an online calibration to synchronize cameras and sensors, and dynamically estimate two neglected factors (i.e., rotational center and time delay).

3. Practical Issues in using Inertial Sensors

In this section, we first describe how to estimate camera motion and blur kernels from inertial sensor readings. This estimation, however, is complicated by a few issues that arise in practice and generally lead to inaccurate results. We then discuss and provide an analysis of these issues.

3.1. Camera Motion Blur

The spatially-invariant image blurring process can be expressed as

$$B = I * k + n, \tag{1}$$

where B is the input blurred image, I is the latent sharp image, $*$ is the convolution operator, k is the blur kernel (point spread function), and n is the noise term. In practice, camera shake usually results in spatially-variant blur determined by the underlying 6D camera motion, including rotations and translations. Inertial sensors, such as gyroscopes and accelerometers, record this information and

have become standard components in smartphones. From their measurements of angular motions and accelerations, camera rotations and translations can be derived and used to estimate spatially-variant blur.

Gyroscope. A gyroscope provides the rate of change in angular position $\theta_R(t)$ with respect to time, i.e., the angular velocity $\alpha(t) = \frac{d\theta_R(t)}{dt}$ in three dimensions with units of deg/s. From these measurements, the angular position can be determined by integrating the angular velocity,

$$\theta_R(t) = \int_0^t \alpha(t)dt \approx \sum_0^t \alpha(t)t_s \tag{2}$$

where t_s is the sampling interval and we assume $\theta_R = 0$ at $t = 0$. Since the camera is a rigid body, each point on the camera shares the same rotation with respect to a certain rotation center $o = (o_x, o_y, o_z)$. The kernel at each location is thus dependent on the rotation center.

Accelerometer. We assume that the change of acceleration, $a'_s(t)$, is measured by the accelerometer in three dimensions with respect to time. With $a_s(t)$ denoting the measured acceleration, we thus have $a'_s(t) = \frac{da_s(t)}{dt}$. Similar to [15], we model the actual acceleration $a(t)$ as

$$a_s(t) = a(t) + g + (R_\theta(t) \times (R_\theta(t) \times r(t))) + (\alpha(t) \times r(t)), \tag{3}$$

where $r(t)$ represents the vector from the accelerometer to the center of rotation, o, and $g(t)$ is the gravity expressed in the current coordinate system. As indicated by this model, the measured acceleration is affected by the actual acceleration, gravity g, centripetal acceleration caused by rotation, and the tangential component of angular acceleration. The position of the accelerometer relative to its initial point at $t = 0$ can be expressed in terms of its initial speed v_0 as

$$\theta_T(t) = \int_0^t v(t)dt = \int_0^t v_0 + (a_s(t) - g)tdt \tag{4}$$

$$\approx \sum_0^t v_0 t_s + 1/2 * (a_s(t) - g)t_s^2. \tag{5}$$

The estimated angular position θ_R and translational position θ_T are taken as the camera pose $\theta = (\theta_R, \theta_T)$ for calculating the kernel at each location.

3.2. Practical Issues

Previous work [15, 24] have made great progress in incorporating inertial sensor data in the deblurring process. These techniques, however, assume that the sensor readings are reliable and that the model used to generate blur kernels accounts for all the contributing factors. Here we present the practical issues involved in using the inertial sensors in smartphones.

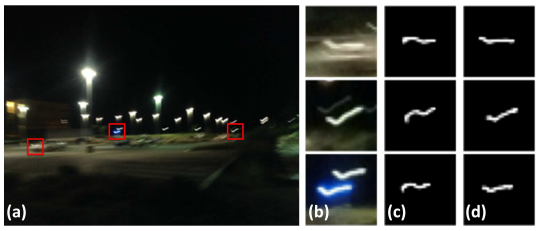


Figure 2. Problem of sensor synchronization. (a) Input blurry image. (b) Light streak patches from blurry image. (c) Generated PSFs using inertial sensor data before synchronization. (d) Generated PSFs after synchronization by our method.

Time Delay. It is well known that the inertial sensors and camera sensor are not well synchronized in smartphones, due to differences in warm-up times. The sensors in our experiment (from an iPhone 6) support a sampling rate of 100 Hz, which is equivalent to a sampling interval of $t_s = 10$ ms. In tests on 20 examples, we found variations in time delay from 10 ms to 120 ms, which corresponds to 1 to 12 data samples from the inertial sensors. We additionally found that estimating the correct time delay is critical to the quality of deblurring, and this variable time delay cannot be determined by offline calibration. To tackle this issue, an online calibration method is proposed in Section 4.1. An example of its effect is shown in Figure 2.

Rotation Center. It is often assumed that the rotation center lies at the optical center of the camera [15, 24]. In reality, the center of rotation may be located at a point far away, such as in the photographer's wrist if the camera motion is caused by hand shake. This discrepancy in the rotational center can significantly affect the computation of blur kernels, as discussed in [19]. Park et al. [19] use multiple shots with the same exposure time to jointly estimate the rotational center, sensor drift and time delay. Their method assumes that the rotation center does not change when taking a sequence of images, which is not necessarily the case. In our tests, we found that the rotation center may even vary during capture. For a single shot with a not very long exposure, the change in rotation center is generally slight and can be neglected. But for different shots, the rotation centers are often varied and this would result in inaccurate motion estimation if the rotation center is assumed to be unchanged. Our single image approach addresses this challenging issue through an online calibration method, as described in Section 4.1. The effect of this online calibration is illustrated in Figure 3.

Noisy Sensor Data. Since inertial sensors in smartphones are not intended for high-quality deblurring, the recorded sensor data are typically noisy, especially for accelerometers. We have examined the noise on our smartphone with the sensors held stationary, i.e., with zero angular acceleration and a constant acceleration from only gravity. We

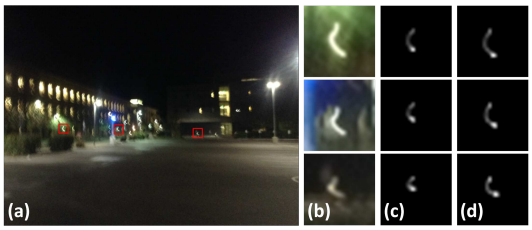


Figure 3. Effect of rotational center estimation. (a) Blurry input image. (b) Patches from the blurry image. (c) PSFs computed using the optical center as the rotational center. (d) PSFs generated using the rotational center estimated by our method.

found that the gyroscope data has relatively less noise, while the accelerometer data exhibits considerable noise, as sensor drifts arise from the integral over the acceleration data in (5). To approximate the drift model from noisy data, a naive idea is to fit a linear function to the resulting deviation of the stationary device. However, the noise with each captured image appears to be different and cannot be adequately recovered with a generic linear model. Instead, we propose an image-based method to refine the blur kernels estimated from sensor data in Section 4.2.

Gravity Effect. As described in Section 3.1, the contribution of gravity to the measured translation $a_s(t)$ is unknown, since it is difficult to estimate the initial orientation and velocity of the camera. If the camera is in an upright position, then gravity would mainly affect y-axis translation in the camera's local coordinate system. If the camera is oriented instead with a 90^o rotation, then the gravity would mainly affect the x-axis translation. This problem can be partially addressed by taking the mean of the translation measurements from the accelerometer [15] when the camera has little translation and is in an upright position. To address this issue better, we consider an image-based method (described in Section 4.2) as well. The errors from both noisy sensor data and the gravity effect can be regarded as the noise term in our kernel estimation.

Scene Depth. Scene depth is a critical element for estimating spatially-variant blur, and its effects always appear when there are camera translations. However, scene depth is usually unknown and difficult to infer from a single image. Assuming a constant depth for the whole scene would introduce errors to kernels at locations with large depth differences. In this work, we take advantage of the phase-based auto-focus built into most popular smartphones to provide sparse depth information, which can be helpful to verify and refine our kernel estimation. The details will be described in Section 4.2.

The aforementioned issues are usually neglected in previous work and cannot be well addressed solely using sensor readings. We therefore advocate using image data to help calibrate camera/sensor parameters online and to reduce errors in blur kernel estimation. The joint use of image and sensor data leads to better performance than using either one alone.

4. Deblurring System

Based on the previous analysis, we propose a deblurring system using the data from inertial sensors in smartphones. Our test platform is an iPhone 6 with iOS 8. The device is equipped with a 8MP camera, gyroscope and accelerometer. Our data capturing app supports adjustments of exposure time and focus, and our deblurring framework consists of online calibration, kernel estimation and refinement, and non-blind image deconvolution.

4.1. Online Calibration

In this work, we propose an image-based online calibration approach to estimate the time delay and rotational center. Whenever available, we specifically take advantage of light streaks, a commonly existing phenomenon first exploited by Hu et al. [13]. Light streaks are imaged paths of point light sources, which indicate the PSFs at the corresponding locations and thus provide reliable shape cues for estimating blur kernels. We adopt the method in [13] to extract light streaks and use them as visual cues in the calibration. We denote a set of detected light streaks (flipped to match the blur kernel) as $\{l_i\}_i$, and we use $k^{(i)}(t_s, o)$ to represent the corresponding kernel at location i with respect to time delay t_s and rotational center o. The exact time delay is adjusted according to the kernel location i and the rolling shutter effect. We interpolate the sensor data by a factor of 4 to account for the higher sampling rate of the rolling shutter effect. The calibration process can thus be formulated as

$$(\hat{t}_s, \hat{o}) = \arg\min_{t_s,o} \sum_i ||k^{(i)}(t_s, o) - l_i||^2. \quad (6)$$

With the rotational center assumed to be constant during the exposure time and located in the imaging plane, i.e. $o_z = 0$, we need only to estimate o_x and o_y. The parameters are estimated in a coarse-to-fine manner.

If no light streaks are detected in an image, we instead make use of the power spectrum analysis presented in [7] and reformulate the calibration in (6) as

$$(\hat{t}_s, \hat{o}) = \arg\min_{t_s,o} \sum_i ||P(k^{(i)}(t_s, o)) - P(p_i * \mathcal{L})||^2, \quad (7)$$

where P is the spectral autocorrelation, and $\mathcal{L}$ is the square root of the Laplacian filter (please see [7] for details). Here, p_i denotes a region that covers a significant number of salient edges determined from the magnitude of image gradients. Since we consider spatially-varying blur in this work, the region size of p_i cannot be large, and it is set to twice the kernel support size in our implementation.

Since noise and saturation in light streaks and patches may reduce accuracy, we fine-tune the parameters by searching within a neighborhood of their values estimated from (6), and identify the solution with the lowest blur level in the results. This is done by first extracting a few non-saturated regions $\{r_j\}_j$ similar to $\{p_i\}_i$ that contain salient edges but are of a larger size. For each region, we generate its kernel using the estimated parameters and apply non-blind deconvolution to obtain the deblurred results. Determining the blur level is a challenging task that has been addressed in recent algorithms [18, 23] using measures based on image statistics and learning techniques. In our task, there exists a true pair of parameter settings with which the correct kernels can be approximately derived from the sensor readings. Thus, we propose a measure on sharpness to identify the correct setting.

According to [12, 7], the spectral autocorrelation of image derivatives in a sharp image is close to that of a delta function. We use this property to fine-tune the estimates for (t_s, o) as

$$(\bar{t}_s, \bar{o}) = \arg \max_{(t_s,o)\in\mathcal{N}(\hat{t}_s,\hat{o})} \sum_j (w_j R(\nabla I_j(t_s, o))), \quad (8)$$

where R represents the maximum response of the autocorrelation function, ∇I_j is the deblurred patch corresponding to r_j using the kernels generated by the parameter setting (t_s, o), and w_j are weights that give greater emphasis to patches with more texture. For deconvolution, we employ the Richardson-Lucy (RL) method due to its simplicity. Moreover, since the RL method acts as an inverse filter, the sum of the intensity values does not change much. In Figure 3, we display an example of kernels recovered using our estimated rotational center, which are seen to be more accurate than those computed using the optical center as the rotational center. In this example, we minimize the influence of translational blur by deliberately avoiding translations in our hand shake and capturing a distant scene.

4.2. Blur Kernel Refinement

The cumulative errors from sensor noise, scene depth, and calibration error can lead to inaccurately estimated kernels from sensor readings. We consider the errors in the noise term and propose to refine the blur kernels in a non-uniform deblurring framework. Similar to [11], we employ a region-based non-uniform blur model that assumes a locally uniform kernel within each region:

$$B = \sum_r k^{(r)} * (\omega^{(r)} \odot I) + n, \quad (9)$$

where r is a region index, and $\omega^{(r)}$ is an image of weights with the same dimensions as the latent image I, such that the pixels in region r can be expressed as $\omega^{(r)} \odot I$ with $\odot$ denoting the pixel-wise product. Here, $k^{(r)}$ represents the kernel at region r and is modeled as a linear combination of kernel bases b:

$$k^{(r)} = \sum_\theta \mu_\theta b_\theta^{(r)}. \quad (10)$$

Each kernel basis b_θ is induced by the homography H_θ of a sampled camera pose θ [11]:

$$b_\theta = C(R_\theta + \frac{1}{d} T_\theta [0, 0, 1])C^{-1} \quad (11)$$

where R_θ and T_θ are the rotation matrix and translation vector for pose θ. C denotes the intrinsic matrix of the camera, which needs to be calibrated beforehand. d represents scene depth. The variables μ are the coefficients of the kernel bases, which are determined by the rotational and translational camera movement.

To remove the blur and recover the latent image, we solve an optimization problem for both the latent image I and the coefficients μ:

$$\arg\min_{I,\mu} \| \sum_r \sum_\theta \mu_\theta b_\theta^{(r)} * (\omega^{(r)} \odot I) - B\|^2 + \beta\varphi(I) + \gamma\varphi(\mu), \quad (12)$$

where $\varphi(I)$ and $\varphi(\mu)$ are regularization terms for the latent image I and the coefficients μ. The two regularization terms are weighted respectively by the parameters β and γ.

The refinement is performed in a coarse-to-fine manner, with the kernels estimated from the sensor readings as initialization. For efficiency, the kernels are refined for each region r_j. Using the geometric model in (12), we alternatingly estimate the coefficients μ:

$$\arg\min_{\mu} \| \sum_\theta \mu_\theta (\sum_{r_j} b_\theta^{(r_j)} T^{(r_j)}) - B^{(r_j)}\|^2 + \gamma\varphi(\mu), \quad (13)$$

and regions of the latent image $T^{(r_j)} = \omega^{(r)} \odot I$:

$$\arg\min_{T^{(r_j)}} \| (\sum_\theta \mu_\theta b_\theta^{(r_j)}) T^{(r_j)} - B^{(r_j)}\|^2 + \beta\varphi(T^{(r_j)}), \quad (14)$$

where $B^{(r_j)}$ and $I^{(r_j)}$ represent corresponding regions in the blurry image and latent image, respectively. The regularization term $\varphi(T^{(r_j)})$ is defined as $\varphi(T^{(r_j)}) = \partial_x T^{(r_j)} + \partial_y T^{(r_j)}$, and has been shown to be effective in recovering smooth results in recent work [22, 4]. We set the term $\varphi(\mu)$ so that the coefficients μ are similar to the coefficients upsampled from the coarser level μ_c: $\varphi(\mu) = ||\mu - (\mu_c)\uparrow||^2$, where $\uparrow$ denotes bilinear upsampling in the camera pose space. The initial coefficients at the coarsest level are obtained by downsampling the values estimated from the inertial sensors. We note that the coarse-to-fine scheme cannot have too many levels, otherwise the kernels estimated from the inertial sensors would have little effect on the optimization at the finest level. The number of levels should depend

on the accuracy of the initial estimation. In this work, we use three levels with a downsampling factor of $\sqrt{2}$.

The kernel basis is built using angular position θ_R and translational position θ_T from sensor readings in (11). Here, to get the depth information, we make use of the built-in phase-detection based auto-focus that is equipped in mobile devices such as the iPhone and Samsung Galaxy. With the help of this component, we are able to obtain a factor value related to depth at each focus point sampled on a grid prior to image capture. In this work, we directly extract the factor values using the iOS API, and calibrate its correspondence to exact scene depth by shooting objects at known depths. In the experiments, we use 6×9 overlapping tiles discussed in (9), and depth values are sampled at the centers of the regions. Each region is associated with a depth value obtained using our data capture app. The depth at each site is then used to determine the blur kernel using (10) and (11).

4.3. Non-Blind Deconvolution

Once we estimate the coefficients μ, we synthesize the blur kernels using (10) and apply non-blind deconvolution to recover the latent sharp image. For the non-blind deconvolution, we employ the method of [28], which is able to handle outliers such as saturation which often exist in low-light images. To better handle noise in the blurry image, the method in [33] could be used.

5. Experimental Results

In this section, we show the results of our sensor-based deblurring method on downsampled images of resolution 800×600. We evaluate each step of our method and present results on real blurry images.

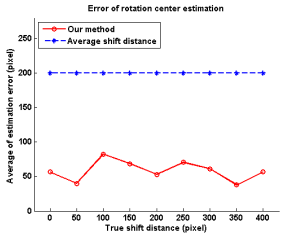
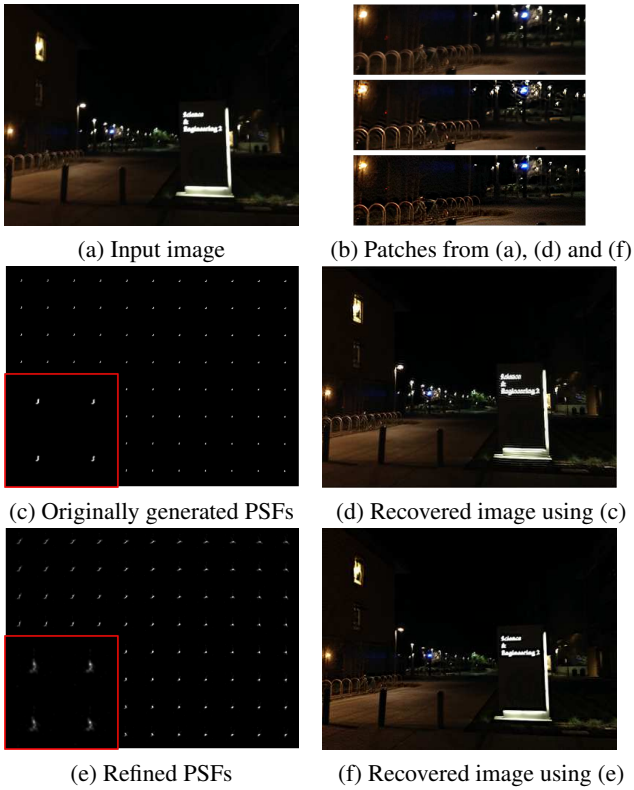


Figure 4. Rotation center estimation with respect to shift distance on synthetic examples. x-axis: distance between the true rotation center and the optical center; y-axis: average distance between estimated rotation centers and the optical center.

5.1. Calibration

To validate our proposed method in estimating the time delay t_s and rotational center o, we test the optimization of (6) as well as the measure in (8) on synthetic examples.

For evaluating the rotational center estimation, we synthesize blurry images with selected rotation centers, and



(a) Input image (b) Patches from (a), (d) and (f)

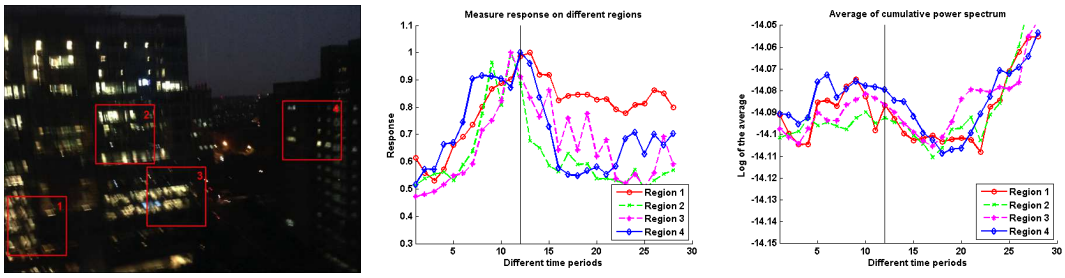
(c) Originally generated PSFs (d) Recovered image using (c)

(e) Refined PSFs (f) Recovered image using (e)

Figure 6. Examples of PSF refinement with inaccurate sensor readings. (c)(e) Zoom-in views are shown at the bottom-left corners.

with camera motions of data sequences collected from inertial sensors. The dataset consists of 2 sharp images, 5 sets of sensor readings and 9 selected rotation centers shifted from the optical center by $0, 50, \ldots, 400$ pixels, for a total of 90 blurry test images. We present the errors with respect to shift distance in Figure 4. As shown in the figure, the proposed method performs consistently well in terms of shift distance, with the estimated rotation centers slightly displaced from the correct values.

To evaluate the autocorrelation-based measure of image sharpness in (8), we test it on four regions of a synthetic image. Blurry images are synthesized from it by applying the blurs for different camera motions taken from the inertial sensors at different time periods. Figure 5(b) depicts the normalized response curve for each region with respect to time period, where each time period is a forward shift from the previous time period by 10 ms (one sample period of the inertial sensors). Here, the x-axis represents the time delay with respect to the number of samples. The maximum responses of the curves occur at time periods near to the correct period. An alternative measure based on the average of the cumulative power spectrum was proposed in [23], where the measure is smaller for a blurry patch than for a sharp patch. We apply the measure of [23] on



(a) (b) (c)

Figure 5. Measure from (8) for identifying the correct parameters. (a) Input blurry image and selected regions. (b) Responses from our method on different regions. (c) Average of the cumulative power spectrum [23] on different regions. The black lines in the figures indicate the correct time periods.

the deblurred regions as well and present the results in Figure 5(c). The results show that our method performs more consistently in this task, and the measure of [23] produces similar values for different periods. The reason is that the kernels generated from these time periods are similar and therefore lead to similar deblurred results, which are not distinguished well using the cumulative power spectrum in comparison to our method based on the maximum spectral autocorrelation response.

5.2. Kernel Refinement

We also validate the kernel refinement step using blurry images synthesized from camera motions computed from the data sequences of the inertial sensors. Noise with the standard deviation of the sensor readings is added to simulate our inaccurate sensor inputs, which are used to compute our kernel initialization. We perform tests on 20 synthetic examples and obtain an average PSNR of 26.62 after refinement, in comparison to 25.05 before refinement. Figure 6 shows an example of our kernel refinement. With the refinement step, more accurate blur kernels are estimated, leading to better deblurring results.

5.3. Deblurring Results

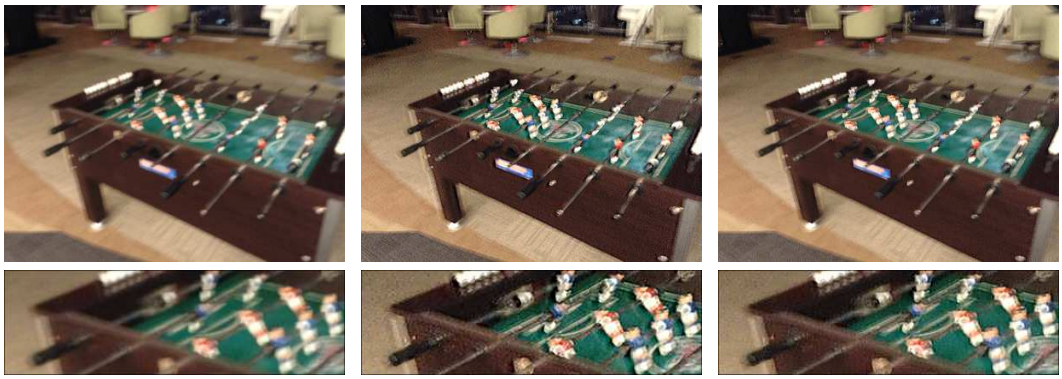
Here, we qualitatively compare our method to state-of-the-art general image deblurring techniques [4, 30, 16, 7], the method based on light streaks [13], and a non-uniform deblurring method [28] on some real-world examples. As shown in Figure 7, previous single-image deblurring methods that do not utilize inertial sensor data perform poorly on these images. For images with insufficient salient edges for kernel estimation, they are likely to converge to a delta kernel. We also compare to results without the rotational center estimation and kernel refinement step, which is similar to previous sensor-based methods [15, 24]. In Figure 7(a)-(f), our method obtains results of higher quality than other methods, though our results exhibit some ringing artifacts. This is because the varying depth values of the foosball table degrade kernel estimation. The method in [13] fails in situations where light streaks cannot be well extracted. In Figure 7(g)-(l), the results from our method without the kernel refinement step contain some blurs, while those after kernel refinement are sharper, which also demonstrates the effectiveness of the refinement process.

6. Discussion

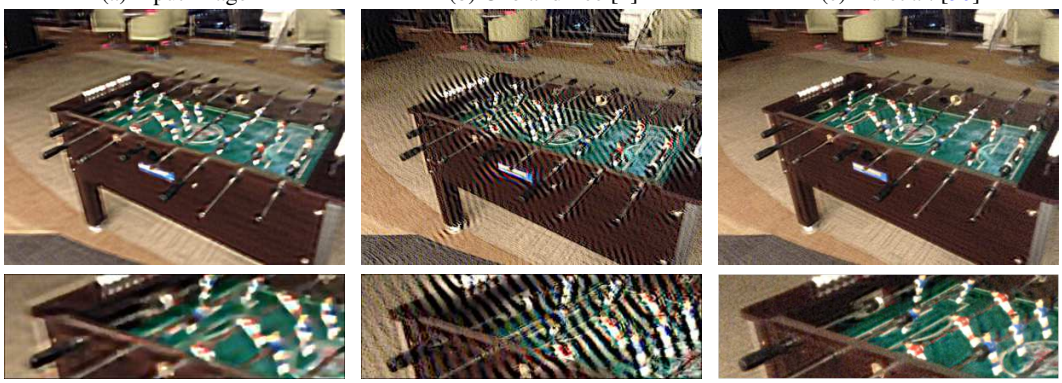
In this work, we examine the problems that arise in image deblurring with smartphone inertial sensors and propose a single-image deblurring method to address these practical issues. A central element of our method is an image-based technique for calibrating the sensors online and refining the inaccurate spatially-variant blur kernels.

There are a few limitations of the proposed method. One is that the method is unable to handle cases when the blur kernels initially estimated from the sensor readings differ substantially from the true kernels. This may occur when the scene depth is shallow, e.g. less than 0.5 meter, and the camera undergoes significant translational motion. Since our kernel refinement method is initialized by the kernels estimated from the sensor readings, poor initial kernels may bias the final results. Another limitation is the amount of computation. Since our method employs a refinement step involving a joint optimization of the non-uniform blur model, the computational load is heavier than that of simply applying non-blind deconvolution with the kernels directly reconstructed from the sensor. A possible solution is to employ cloud processing by uploading the blurred image and sensor readings. How to make the computation efficient on mobile devices is an interesting direction for future work. Moreover, we only obtain depth values at sparsely sampled points and correspond them to tiled regions. To obtain dense depth estimation, a multiple camera system, e.g., the Amazon Fire phone, could instead be used with stereo algorithms.

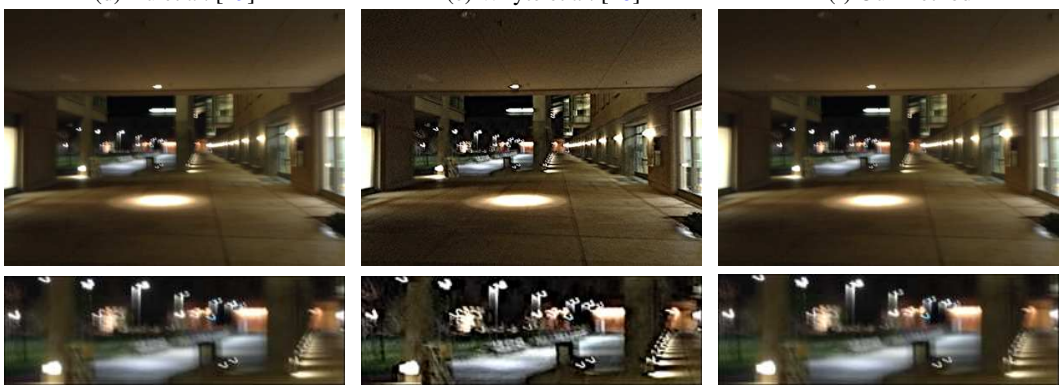
Acknowledgements: Z. Hu and M.-H. Yang are supported in part by the NSF CAREER Grant (No. 1149783), NSF IIS Grant (No. 1152576).



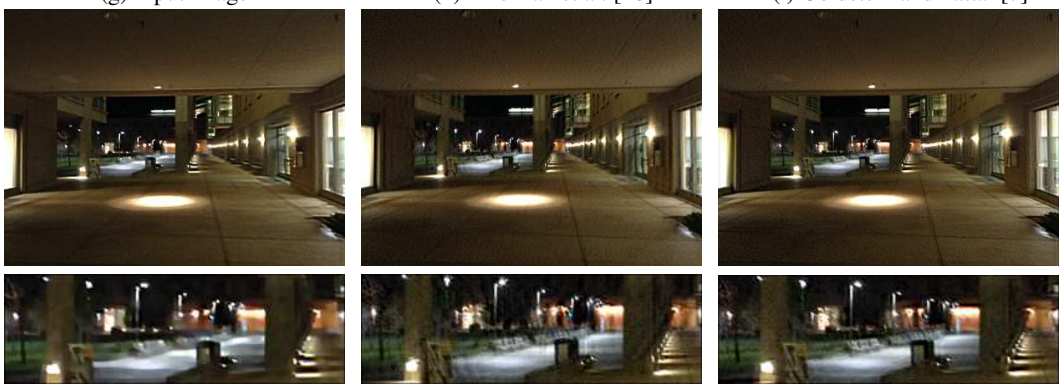
(a) Input image (b) Cho and Lee [4] (c) Xu et al. [30]



(d) Hu et al. [13] (e) Whyte et al. [28] (f) Our method



(g) Input image (h) Krishnan et al. [16] (i) Goldstein and Fattal [7]



(j) Hu et al. [13] (k) Ours without kernel refinement (l) Our method

Figure 7. Comparisons on real examples. The results are best viewed on a high-resolution digital display.

References

[1] M. Ben-Ezra and S. Nayar. Motion-based motion deblurring. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):689–698, June 2004. 2

[2] J. Cai, H. Ji, C. Liu, and Z. Shen. Blind motion deblurring from a single image using sparse approximation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 2

[3] J. Chen, L. Yuan, C.-K. Tang, and L. Quan. Robust dual motion deblurring. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2008. 2

[4] S. Cho and S. Lee. Fast motion deblurring. In *Proceedings of ACM SIGGRAPH Asia*, 2009. 5, 7, 8

[5] T. S. Cho, N. Joshi, C. L. Zitnick, S. B. Kang, R. Szeliski, and W. T. Freeman. A content-aware image prior. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 2

[6] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. In *Proceedings of ACM SIGGRAPH*, 2006. 2

[7] A. Goldstein and R. Fattal. Blur-kernel estimation from spectral irregularities. In *Proceedings of European Conference on Computer Vision*, 2012. 4, 5, 7, 8

[8] A. Gupta, N. Joshi, L. Zitnick, M. Cohen, and B. Curless. Single image deblurring using motion density functions. In *Proceedings of European Conference on Computer Vision*, 2010. 2

[9] Y. Hacohen, E. Shechtman, and D. Lischinski. Deblurring by example using dense correspondence. In *Proceedings of IEEE International Conference on Computer Vision*, 2013. 2

[10] S. Harmeling, M. Hirsch, and B. Schölkopf. Space-variant single-image blind deconvolution for removing camera shake. In *Neural Information Processing Systems*, 2010. 2

[11] M. Hirsch, C. J. Schuler, S. Harmeling, and B. Schölkopf. Fast removal of non-uniform camera shake. In *Proceedings of IEEE International Conference on Computer Vision*, 2011. 2, 5

[12] W. Hu, J. Xue, and N. Zheng. Psf estimation via gradient domain correlation. *IEEE Transactions on Image Processing*, 21(1):386–392, 2012. 5

[13] Z. Hu, S. Cho, J. Wang, and M.-H. Yang. Deblurring low-light images with light streaks. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 4, 7, 8

[14] J. Jia. Single image motion deblurring using transparency. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2007. 2

[15] N. Joshi, S. B. Kang, C. L. Zitnick, and R. Szeliski. Image deblurring using inertial measurement sensors. In *Proceedings of ACM SIGGRAPH*, 2010. 1, 2, 3, 4, 7

[16] D. Krishnan, T. Tay, and R. Fergus. Blind deconvolution using a normalized sparsity measure. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 7, 8

[17] A. Levin, R. Fergus, F. Durand, and W. T. Freeman. Image and depth from a conventional camera with a coded aperture. In *ACM Transactions on Graphics*, volume 26, 2007. 2

[18] Y. Liu, J. Wang, S. Cho, A. Finkelstein, and S. Rusinkiewicz. A no-reference metric for evaluating the quality of motion deblurring. *ACM Transactions on Graphics*, 32(6), 2013. 5

[19] S. H. Park and M. Levoy. Gyro-based multi-image deconvolution for removing handshake blur. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 1, 2, 3

[20] D. Perrone and P. Favaro. Total variation blind deconvolution: The devil is in the details. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 2

[21] R. Raskar, A. Agrawal, and J. Tumblin. Coded exposure photography: motion deblurring using fluttered shutter. In *Proceedings of ACM SIGGRAPH*, 2006. 2

[22] Q. Shan, J. Jia, and A. Agarwala. High-quality motion deblurring from a single image. In *Proceedings of ACM SIGGRAPH*, 2008. 2, 5

[23] J. Shi, L. Xu, and J. Jia. Discriminative blur detection features. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 5, 6, 7

[24] O. Šindelář and F. Šroubek. Image deblurring in smartphone devices using built-in inertial measurement sensors. *Journal of Electronic Imaging*, 22(1):011003–011003, 2013. 1, 2, 3, 7

[25] L. Sun, S. Cho, J. Wang, and J. Hays. Edge-based blur kernel estimation using patch priors. In *Proceedings of IEEE International Conference on Computational Photography*, 2013. 2

[26] Y. Tai, P. Tan, and M. S. Brown. Richardson-Lucy deblurring for scenes under projective motion path. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1603–1618, 2011. 2

[27] Y.-W. Tai, H. Du, M. Brown, and S. Lin. Correction of spatially varying image and video motion blur using a hybrid camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(6):1012–1028, June 2010. 2

[28] O. Whyte, J. Sivic, and A. Zisserman. Deblurring shaken and partially saturated images. *International Journal Computer Vision*, 110(2):185–201, 2014. 6, 7, 8

[29] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce. Non-uniform deblurring for shaken images. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 2

[30] L. Xu, S. Zheng, and J. Jia. Unnatural l0 sparse representation for natural image deblurring. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2013. 2, 7, 8

[31] L. Yuan, J. Sun, L. Quan, and H.-Y. Shum. Image deblurring with blurred/noisy image pairs. In *ACM Transactions on Graphics*, volume 26. 2

[32] H. Zhang, D. Wipf, and Y. Zhang. Multi-observation blind deconvolution with an adaptive sparse prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8):1628–1643, 2014. 2

[33] L. Zhong, T. S. Cho, D. Metaxas, S. Paris, and J. Wang. Handling noise in single image deblurring using directional filters. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2013. 6