

Image deblurring in smartphone devices using built-in inertial measurement sensors

Ondřej Šindelář

Faculty of Mathematics and Physics,
Charles University in Prague
Email: ondrasindelar@gmail.com

Filip Šroubek

Institute of Information Theory and Automation,
Czech Academy of Science
Pod Vodárenskou věží 4,
CZ-182 08, Praha 8, Czech Republic
Phone: +420266052527
Email: sroubekf@utia.cas.cz

Abstract

Long exposure hand-held photography is degraded with blur, which is difficult to remove without prior information about the camera motion. In this work, we utilize inertial sensors (accelerometers and gyroscopes) in modern smartphones to detect exact motion trajectory of the smartphone camera during exposure and remove blur from the resulting photography based on the recorded motion data. The whole system is implemented on Android platform and embedded in the smartphone device resulting in a close-to-real-time deblurring algorithm. The performance of the proposed system is demonstrated in real-life scenarios.

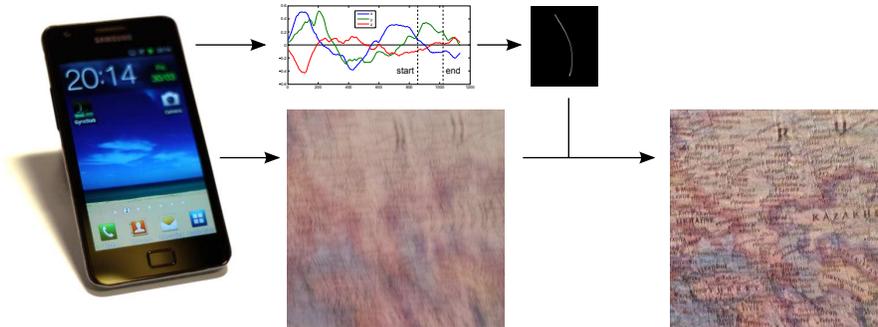


Figure 1: Basic application workflow. Together with a taken photograph gyroscope data are recorded, which is a base for blur kernel estimation. A deconvolution is then performed to remove blur from the image.

1 Introduction

Blur induced by camera motion is a frequent problem in photography mainly when the light conditions are poor. As the exposure time increases, involuntary camera motion has a growing effect on the acquired image. Image stabilization (IS) devices that help to reduce the motion blur by moving the camera sensor in the opposite direction are becoming more common. However, such hardware remedy has its limitations as it can compensate only for motion of a very small extent and speed. Deblurring the image offline using mathematical algorithms is usually the only choice we have in order to obtain a sharp image. Motion blur can be modeled by convolution and the deblurring process is called deconvolution, which is a well-known ill-posed problem. In general, the situation is even more complicated, since we usually have no or limited information about the blur shape.

We can divide the deconvolution methods into two categories: methods that estimate the blur and the sharp image directly from the acquired image (blind

deconvolution) and methods that use information from other sensors to estimate the blur (semi-blind deconvolution).

Over the last few years, blind deconvolution experiences a renaissance. The key idea of new algorithms belonging to the first category is to address the ill-posedness of blind deconvolution by characterizing the image prior using natural image statistics and by a better choice of estimators. A frantic activity started with the work of Fergus *et al.*[2], who applied variational Bayes to approximate the posterior by a simpler distribution. Other authors [5, 6, 12, 18] stick to the “good old” alternating MAP approach, but by using ad hoc steps, which often lack rigorous explanation, they converge to a correct solution. Levin *et al.* in [9, 8] proved that a proper estimator matters more than the shape of priors. They showed that marginalizing the posterior with respect to the latent image leads to the correct solution of the blur. The marginalized probability can be expressed in a closed form only for simple priors that are, e.g., Gaussian. Otherwise approximation methods such as Variational Bayes [11] or the Laplace approximation [3] must be used. Complex camera motion often results in blur that is space-variant, i.e., the blur is a function of a position vector. As a rule, the space-variant blur cannot be expressed by an explicit formula but in many cases it has a special structure that can be exploited. If only one type of camera motion is considered (e.g. rotation), we can express the degradation operator as a linear combination of basis blurs (or images) and solve the blind problem in the space of the basis, which has much lower dimension than the original problem. Whyte *et al.*[17] considered rotations about three axes up to several degrees and described blurring using three basis vectors. For blind deconvolution, they used an algorithm analogous to [2] based on marginalization over the latent sharp image. Gupta *et al.*[4] adopted a similar approach, replacing rotations about x and y axes by translations. State-of-the-art blind deconvolution algorithms

achieve sometimes awesome results. However, their main limitation is that they work only in specific situations, they are prone to local extrema, and they are computationally very demanding.

The second category of deconvolution algorithms (semi-blind) tries to overcome these drawbacks by using information about the camera motion from other sources. One possibility is to acquire a pair of images: one correctly exposed but blurred and one underexposed (noisy) but sharp image. Then we can apply multichannel blind deconvolution methods, which are better posed, as was proposed for example in [16, 19, 14]. Another possibility is to attach an auxiliary high-speed camera of lower resolution to estimate the PSF using for example optical flow techniques [1, 15]. Many devices, such as modern smartphones, are now equipped with inertial sensors (gyroscopes and accelerometers) that can give us a very accurate information about camera motion. If we are able to reconstruct camera path then we can recover blur and perform nonblind image deblurring. This idea was recently described by Joshi *et al.* in [7] but they have designed an expensive measuring apparatus consisting of a DSLR camera and a set of inertial sensors, and perform image deblurring offline on a computer. This work is based on the same idea but our aim is to show that image deblurring is feasible on modern smartphones and not requiring any other devices.

The main contribution of this work is to illustrate that blur estimation with built-in inertial sensors is possible and to implement image deblurring on a smartphone, which works in practical situations and is relatively fast to be acceptable for a general user. The next section shows the relation between the camera pose and the image blur, and discusses simplifications that we make. Sec. 3 briefly describes implementation on our test device (Samsung smartphone). Sec. 4 shows results of our experiments and addresses pitfalls that are common for cameras embedded in smartphones.

2 Camera motion blur analysis

We start the discussion with a general camera motion. Since our primary goal is a handy implementation for mobile devices, we then introduce simplification of the problem that allows a fast and memory-conserving solution with promising results.

2.1 The model

The image degradation model is represented by relation

$$g = H(u) + n, \quad (1)$$

where H is a linear degradation operator and n is additive noise. Image coordinate indices are omitted here for simplicity. Our goal is to find an estimate of the original image u from the observed blurred image g .

To track the effect of camera motion on the output image, we first assume a standard perspective projection $\Pi : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ that transforms a 3D point $[x, y, z]$ in the observed scene to a 2D location $[x', y']$ in the image plane:

$$\Pi([x, y, z]^T) = \left[\frac{xf}{z}, \frac{yf}{z} \right]^T. \quad (2)$$

For the sake of brevity, we assume here only the focal length f in the intrinsic camera matrix. The optical axis is identical with the z axis. During camera motion, projection of a point $p = [x, y, z]^T$ at time τ within the exposure period is given by

$$C(\tau) = \Pi \left(R(\tau) \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} t_x(\tau) \\ t_y(\tau) \\ t_z(\tau) \end{bmatrix} \right) = \Pi(R(\tau)p + t(\tau)), \quad (3)$$

where R and t are 3D rotation matrix and translation vector, respectively, that define the camera pose at time τ . The rotation matrix $R(\tau)$ is given by three rotation angles $\phi_x(\tau)$, $\phi_y(\tau)$ and $\phi_z(\tau)$.

The resulting curve C makes up a trajectory of a trace that is left on the sensor by a point light source. Assuming a constant illuminance over the exposure period, the light energy emitted from the point is distributed evenly (with respect to time) over the curve C . This effectively gives us a time parametrization of a point-spread function (PSF) for a given point, which forms the blur operator H . The operator H can be written in a form naturally generalizing standard convolution as

$$H(u)[x, y] = \int u(x - s, y - t) \tilde{h}(s, t, x - s, y - t) ds dt, \quad (4)$$

where \tilde{h} depends on the position (third and fourth variable) and can be regarded as a space-variant point-spread function.

Now we can draw the relation between \tilde{h} in 4 and the curve C . For any given 3D point at position p rendered on the image plane to $[x', y'] = \Pi(p)$ the point-spread blur function $\tilde{h}(s, t, x', y')$ is a 2D function of $[s, t]$, which can be interpreted as a blurred image of an ideal light point displayed at $[x', y']$. It can be thus obtained by rendering the curve C on a plane with the total integral of \tilde{h} (which has to be equal to 1 to conserve distribution of energy) distributed along the path evenly in respect to the time parameter.

In the next section, we will show how to simplify this model and assume the space-invariant case, i.e. $\tilde{h}(s, t, x, y) = h(s, t)$.

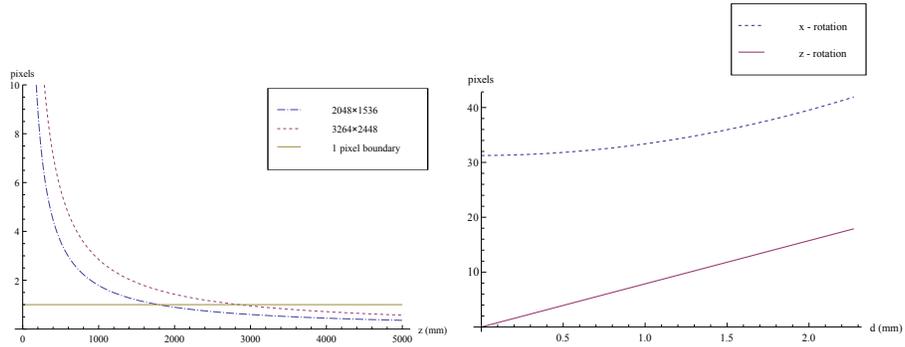
2.2 Space-invariant simplification

We will consider a situation when the operator H is spatially invariant, so (1) becomes

$$g = h * u + n, \quad (5)$$

where “*” denotes convolution and h is a space-invariant PSF.

The PSF formula (3) is spatially variant in general, so it will be modified for our purposes. First of all, the translation t affects the projection differently depending on the object distance from the camera. The relation is inversely proportional, as shown in Fig. 2a. In the case of our test device, if the camera shifts by 1 mm, objects at distance of 2 m or more move by less than 1 pixel in the image. We can thus effectively ignore translation as a cause of blur in many practical situations.



(a) Influence of 1 mm x or y translation depending on object distance. Angle of view is 60° ; two curves represent different image sensor resolution.

(b) Influence of 1° rotation about x and z axis depending on a distance d from the image sensor center. The full sensor extent corresponds to $d = 2.3$ mm; image resolution is 2048×1536 .

Figure 2: Dependence of projection shift on translation and z -rotation for a test device.

Rotation about the optical z axis (yaw) intuitively interferes with the space-invariant blur assumption. This type of rotation applied on a point light source placed in the center of the picture (on the optical axis) leaves the projection

unchanged, but points outside the center form arc-shaped traces that grow towards the image borders. Provided that the camera is rotated with an equal amount around all three axes, which is a fair assumption under normal circumstances, a yaw has the least effect on the resulting blur, especially in the center of the sensor. The cellphone cameras typically have the focal length close to the sensor size, which means that only close to the image borders the blur size produced by yaw is approaching the blur size produced by rotation about x or y ; see 2b.

The last obstacle towards the space-invariant PSF is the perspective projection itself. Length of a trace caused by x and y rotations are projected slightly differently depending on the distance from the optical center, because the rectilinear projection (2) casts a point at an angle α from the optical axis to a point at a distance of $f \cdot \tan(\alpha)$ from the image center. The tangent function is close to linear for small angles, so both x and y rotations by a small angle α shift a point in the sensor center approximately $f \cdot \alpha$ away in the direction of the given axis. Using the same rule for all points on the sensor gives us the space-invariant simplification of (3):

$$C(\tau) \approx \begin{bmatrix} x' \\ y' \end{bmatrix} + f \begin{bmatrix} \phi_x(\tau) \\ \phi_y(\tau) \end{bmatrix}, \quad (6)$$

where $[x', y']$ is the location of a point in the image. This approximation holds if z is large, and $x' \phi_x \ll f$ and $y' \phi_y \ll f$, which is true at least in the central part of the image.

3 Implementation

As a testing platform, we have chosen a *Samsung Galaxy S II* smartphone with Android OS. It is equipped with all the apparatus needed for our experiments, namely a relatively high-quality camera, motion sensors, a fast CPU and enough RAM to perform computations.

3.1 PSF estimation

During the photo acquisition, samples of angular velocity are recorded using the embedded gyroscopes, which are afterwards trimmed to fit the exposure period. An estimation of the PSF is rendered by integrating the curve position from the recorded data using (6).

3.2 Deconvolution

State-of-the-art non-blind deconvolution methods use sparse image priors and the solution is usually found by some iterative minimization algorithms, such as in [12]. However the limited computational power of the smartphone prevents us to implement these sophisticated deconvolution methods. We thus use a simple but fast *Wiener filter* in the form

$$\hat{U} = G \frac{H^*}{|H|^2 + \Phi}, \quad (7)$$

where Φ is an estimation of the inverse Signal to Noise Ratio, and G , H and \hat{U} are discrete Fourier transforms of the observed image g , PSF h and the estimated latent image \hat{u} , respectively.

Filtering in the frequency domain treats the image as a periodic function, which causes ringing artifacts around image borders. To overcome this problem, several less or more sophisticated techniques were proposed in the literature

[10, 13]. We have found sufficient to preprocess the input image g by blending the opposite image borders at the width of the PSF, which creates a smooth transition and eliminates the artifacts.

The intensity values of the output image \hat{u} sometimes lie outside the 8-bit range (0-255), therefore we added optional normalization with clipping of outliers. The normalization is especially useful in the case of larger blurs and scene with high illumination.

For conversions of the images to frequency domain and back, we use FFT algorithm implemented in the FFTW library. Utilizing a fast ARM Cortex-A9 CPU with two cores and support for a SIMD instruction set (NEON), FFTW proved to be remarkably fast on the tested smartphone; see Tab. 1.

resolution	no NEON, no hardware FPU	NEON, 1 core	NEON, 2 cores
1536×1152	2900	185	110
2048×1536	5300	330	195
2050×1538	—	1000	540
3264×2448	21200	1450	800

Table 1: Speed (in milliseconds) of FFT transform of gray-scale images with different sizes and different CPU settings.

The acquired images with native camera resolution of 3264×2448 is by default scaled down to 2048×1536 to take the advantage of better performance of FFTW when the image size is a factor of small primes. Image downsampling has a negligible effect on the image quality, because native camera resolution is unnecessarily high. The optical system of the camera has a very small aperture, which, because of diffraction and optical aberrations, limits the number of pixels that can be effectively captured by the image sensor.

To perform Wiener filtering, FFT must be applied several times: once for the PSF and twice (forward and backward-inverse) for each color channel. That yields a total of 7 FFT operations. With some overhead of bitmap transfers,

the deconvolution phase for the image resolution 2048×1536 takes about 2.6 seconds. The whole process starting from the camera shutter is done in a little over 6 seconds. This includes image resizing, PSF estimation, compressing and saving the original and deblurred image files.

4 Results

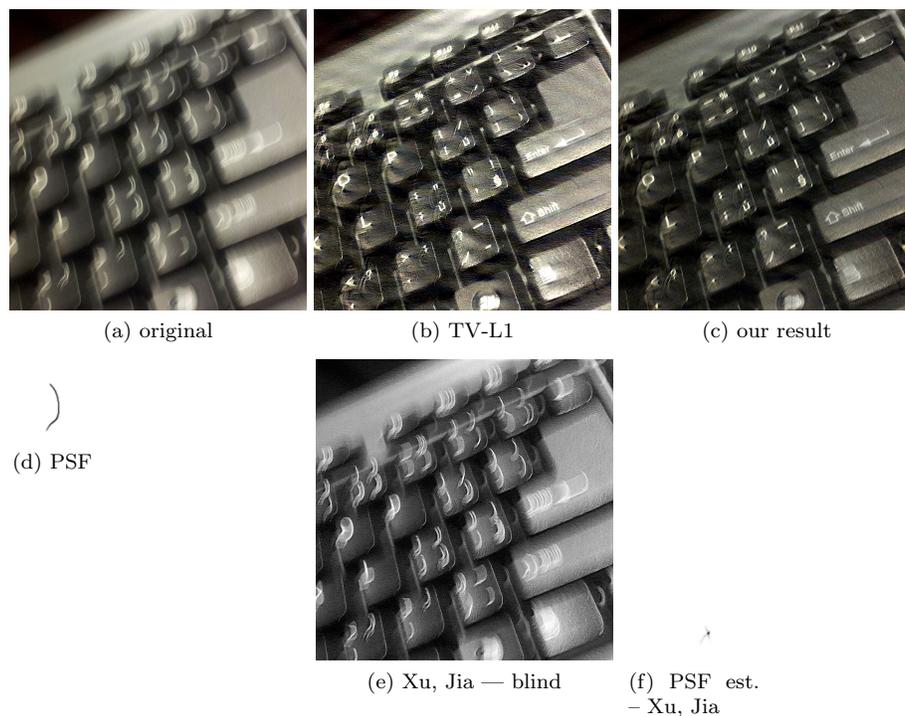


Figure 3: Test 1 — 1/7 s exposure, 16×59 estimated PSF.

In this section we display several of our results together with estimated PSFs; see Figs. 3, 4, 5. All results were computed with the signal-to-noise parameter Φ set to 0.01. This value was determined experimentally to provide the best looking results. The original intention was to set Φ proportionally to ISO value extracted from EXIF data of a photo, which should determine the amount of

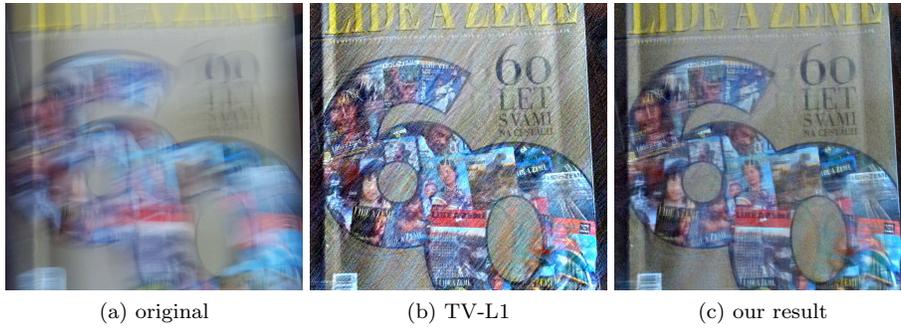


Figure 4: Test 2 — 1/9 s exposure, 21×28 estimated PSF.



Figure 5: Test 3 — 1/2 s exposure, 72×76 estimated PSF.

noise present in the image. However, we found the dependency of Φ on ISO very negligible. We explain this behavior by the denoising step that the mobile phone internally performs on the captured photos.

For comparison, we show an advanced non-blind iterative method (TV-L1)

by Xu and Jia [18]¹, which minimizes image Total Variation and data term in the L_1 -norm. We also tested blind deconvolution proposed in the same, which is probably currently the best blind deconvolution method. However, the result of the first test image shown in Fig. 3e illustrates a total failure of this method when applied to images taken by our test device. The PSF (Fig. 3f) estimated by the blind deconvolution method is close to a delta function and the estimated image (Fig. 3e) is thus a slightly sharpened image. We suspect that small PSF variations in space and/or the image post-processing done by the smartphone prevents a successful estimation of the correct motion blur. The same unsatisfactory behavior was observed in all our tests. However, our results (c) illustrate that in spite of a relatively simple approach that incorporates the Wiener filter with the space-invariant PSF estimated by inertial sensors, the proposed method is capable of producing convincing images exposing many details that were hidden in the original. The non-blind algorithm of Xu and Jia, which is using the same PSF estimated by inertial sensors, tends to amplify the signal, which rather emphasizes noise and false edges than gains signal improvement. Conversely, high frequency details are more suppressed, probably due to being treated as noise, despite of careful attempts to tune the parameters of the method. Within our testing environment, the simplified Wiener filter is more advantageous as it filters all frequencies evenly which apparently matches the spectrum characteristics of most of the tested images.

Our results seem to lack slightly contrast, which is largely because of the normalization. On the other hand, it helps retaining the full dynamic range without saturation as clearly seen in the comparison Fig. 3.

Our deconvolution process admittedly has downsides, as well. Focusing in a dark environment may be unsuccessful and then the deconvolved result cannot

¹An executable is available for download at <http://appsrv.cse.cuhk.edu.hk/~xuli/deconv.zip>

be sharp even if the PSF estimation is correct, since we lack any means to estimate the out-of-focus blur.

The subjective quality of the deconvolution output is not entirely consistent. Images presented in this section are the best-looking results. Outputs of the similar quality are frequently achieved by our method, but sometimes the result is impaired by visual anomalies worsening its appearance. Most often it is manifested as ringing artifacts surrounding sharp edges in the picture, as demonstrated in Fig. 6.

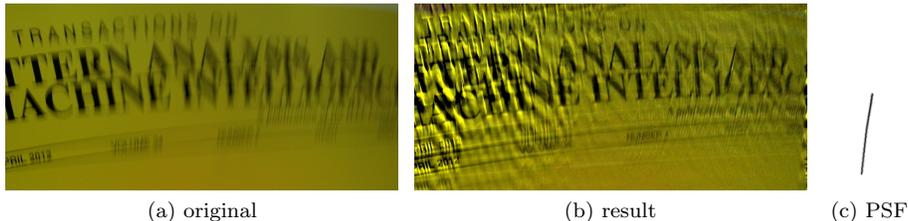


Figure 6: An example of an unsatisfactory result.

The lack of control over camera hardware in the phone (no manual exposure settings, no access to raw data from the image sensor) and inaccurate timing of exposure events prevents us to systematically evaluate our method and find sources of malfunctioning.

The main problem is most likely the space-variant nature of the PSF as discussed in Sec. 2, which is particularly noticeable when a rotation about the z axis is significant or a translation movement is present and the scene depth is small. The example in Fig. 6 is influenced by a combination of both of these factors. The space-invariant approximation of camera projection is often apparent in parts close to image borders, because of a relatively wide camera field of view (60°).

However, another cause is the shutter mechanism. Contrary to systems with

mechanical shutter, values of illuminated pixels are here read successively line by line. The readout from the CMOS sensor takes several tens of milliseconds, which results in a picture not taken at a single moment, but with a slight time delay between the first and last pixel row. This process, called *rolling shutter*, is therefore another cause of the blur variance as the PSF depends on the vertical position in the image. The correct approach to PSF estimation is thus shifting inertial sensor data in time according to the vertical position in the image.

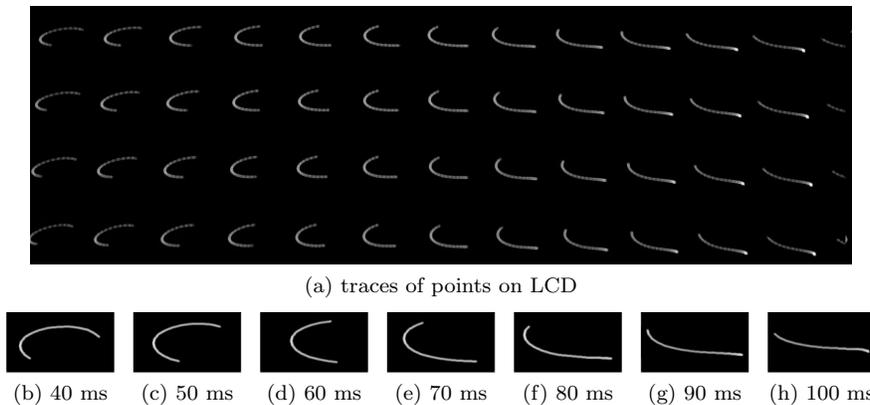


Figure 7: A snapshot of point grid displayed on a LCD screen showing the rolling shutter effect. The bottom row shows a series of blur kernels rendered using data from the gyroscope sensor shifted in time. Exposure $1/14$ s, PSF images were created from sensor data starting 40–100 ms after a synchronization timestamp.

The application programming interface (API) of the tested device does not allow accurate synchronization between camera and gyroscope samples. Therefore we have implemented a deconvolution preview, where the user picks the best option from a set of results created with time-shifted PSFs. The preview also partly solves the rolling shutter problem since the selected time shift corresponds to a horizontal image band of a certain height that can be considered as acquired at one moment, thus eliminating the rolling shutter effect for that image part.

Image post-processing might also present a serious problem for the deconvolution. Since the original raw data from the image sensor are not available, we are forced to work with the JPEG-compressed image, which is most likely processed by a denoising, contrast enhancement algorithm or lens distortion compensation. These adjustments are undesirable for our purposes, as they were not taken into account in our model.



Figure 8: Noise in gyroscope data. Synthetically blurred Lena image using PSF from recorded gyroscope samples and afterwards deblurred using PSF from measurements with variable amount of noise. Images are from left to right, top to bottom: original, blurred and 6 deblurred images using original gyroscope data altered by random gaussian noise with variance from 0 to 0.05 (gyroscope measurements are in rad/s).

Noise present in gyroscope measurement data can also be a problem, as displayed in Figs. 8 and 9. This has been examined in a following synthetic experiment. A test image was first blurred using convolution with a PSF counted from one set of gyroscope samples recorded in our mobile application. An ad-

ditive noise was added to the image in accordance with the model 1 (40 dB Gaussian noise was used). Gaussian noise was also added to the gyroscope samples to simulate errors in sensor measurement. Corrupted image was then repaired using our deblurring algorithm from the altered motion data. Results for different amounts of noise in gyroscope samples are shown in Fig. 8. The mean square error of the result as a function of the gyroscope noise level (variance) is in Fig. 9. We can see that the performance starts to drop for noise levels above 0.05 rad/s. The gyroscope noise level typically encountered in the motion sensors inside mobile devices (in our case Samsung Galaxy S II) is 0.007 rad/s for our sampling rate and it is therefore way below the critical level.

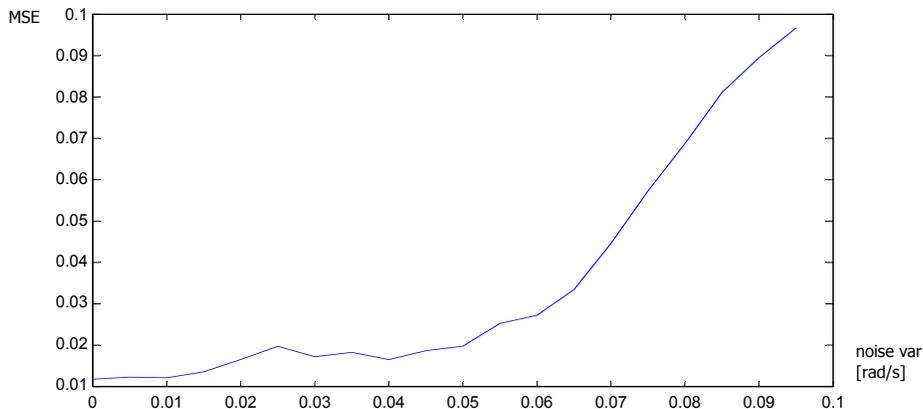


Figure 9: MSE of difference between the original and deblurred image in relation to amount of added sensor noise. Gaussian noise of variance 0 to 0.1 was added to gyroscope measurements (angular velocity in rad/s). Deconvolution algorithm was then performed using computed blur kernels based on these altered measurements. Mean squared error of difference to the original image is plotted in the graph (pixel value was normalized to $\langle 0, 1 \rangle$ range). The graph shows mean of 10 iterations for each of the variance values. Lena image was used for the test.

5 Conclusion

We have presented an image deblurring method that can effectively remove blur caused by camera motion using information from inertial sensors. The proposed method is fully implemented on a smartphone device, which is to our knowledge the first attempt in this direction and renders the method particularly appealing for end users. We have justified the space-invariant simplification for certain camera motions, but simultaneously we have uncovered intrinsic sources of space-variant blur, such as rolling shutter. The space-variant implementation of the deblurring algorithm, which would solve some of the current issues, is in theory possible, but the computational cost on the smartphone may be too high. It will be a topic of our future research to find out whether this is viable.

References

- [1] Moshe Ben-Ezra and Shree K. Nayar. Motion-based motion deblurring. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):689–698, June 2004.
- [2] Rob Fergus, Barun Singh, Aaron Hertzmann, Sam T. Roweis, and William T. Freeman. Removing camera shake from a single photograph. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, pages 787–794, New York, NY, USA, 2006. ACM.
- [3] N. P. Galatsanos, V. Z. Mesarovic, R. Molina, and A. K. Katsaggelos. Hierarchical bayesian image restoration from partially known blurs. *IEEE Transactions on Image Processing*, 9(10):1784–1797, 2000.
- [4] Ankit Gupta, Neel Joshi, C. Lawrence Zitnick, Michael Cohen, and Brian Curless. Single image deblurring using motion density functions. In *Proceedings of the 11th European conference on Computer vision: Part I, ECCV'10*, pages 171–184, Berlin, Heidelberg, 2010. Springer-Verlag.
- [5] Jiaya Jia. Single image motion deblurring using transparency. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition CVPR '07*, pages 1–8, 17–22 June 2007.
- [6] N. Joshi, R. Szeliski, and D. J. Kriegman. PSF estimation using sharp edge prediction. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition CVPR 2008*, pages 1–8, 23–28 June 2008.
- [7] Neel Joshi, Sing Bing Kang, C. Lawrence Zitnick, and Richard Szeliski. Image deblurring using inertial measurement sensors. *ACM Trans. Graph.*, 29:30:1–30:9, July 2010.

- [8] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Understanding blind deconvolution algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(12):2354–2367, 2011.
- [9] A. Levin, Y. Weiss, F. Durand, and W.T. Freeman. Understanding and evaluating blind deconvolution algorithms. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition CVPR '09*, 2009.
- [10] Renting Liu and Jiaya Jia. Reducing boundary artifacts in image deconvolution. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 505 –508, oct. 2008.
- [11] James Miskin and David J.C. MacKay. Ensemble learning for blind image separation and deconvolution. In M. Girolani, editor, *Advances in Independent Component Analysis*. Springer-Verlag, 2000.
- [12] Qi Shan, Jiaya Jia, and Aseem Agarwala. High-quality motion deblurring from a single image. In *SIGGRAPH '08: ACM SIGGRAPH 2008 papers*, pages 1–10, New York, NY, USA, 2008. ACM.
- [13] M. Sorel. Removing boundary artifacts for real-time iterated shrinkage deconvolution. *Image Processing, IEEE Transactions on*, 21(4):2329 –2334, april 2012.
- [14] M. Šorel and F. Šroubek. Space-variant deblurring using one blurred and one underexposed image. In *Proceedings of the IEEE 16th International Conference on Image Processing ICIP 2009*. IEEE, 2009.
- [15] Yu-Wing Tai, Hao Du, Michael S. Brown, and Stephen Lin. Correction of spatially varying image and video motion blur using a hybrid camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:1012–1028, 2010.

- [16] M. Tico, M. Trimeche, and M. Vehvilainen. Motion blur identification based on differently exposed images. In *Proc. IEEE Int. Conf. Image Processing*, pages 2021–2024, 2006.
- [17] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce. Non-uniform deblurring for shaken images. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 491–498, June 2010.
- [18] Li Xu and Jiaya Jia. Two-phase kernel estimation for robust motion deblurring. In *Proceedings of the 11th European conference on Computer vision: Part I, ECCV'10*, pages 157–170, Berlin, Heidelberg, 2010. Springer-Verlag.
- [19] Lu Yuan, Jian Sun, Long Quan, and Heung-Yeung Shum. Image deblurring with blurred/noisy image pairs. In *SIGGRAPH '07: ACM SIGGRAPH 2007 papers*, page 1, New York, NY, USA, 2007. ACM.