https://doi.org/10.2352/ISSN.2470-1173.2020.14.COIMG-379

A Dataset for Deep Image Deblurring Aided by Inertial Sensor Data

Shuang Zhang*, Ada Zhen, Robert L. Stevenson; University of Notre Dame; Notre Dame, Indiana, 46556

Abstract

Recent work in image deblurring aided by inertial sensor data has shown promise. Separate work has also shown that deep learning techniques are useful for the image deblurring problem. Due to a lack of a proper dataset, however, deep learning techniques have not yet to be successfully applied to image deblurring when inertial sensor data is also available. This paper proposes to generate a synthetic training and testing dataset that includes groundtruth and blurry image pairs as well as inertial sensor data recorded during the exposure time of each blurry image. To simulate the real situations, the proposed dataset called DeblurIMUDataset considers synchronization issue, rotation center shift, rolling shutter effect as well as inertial sensor data noise and image noise. This dataset is available online[1].

1. Introduction

Image deblurring is an inverse process that removes motion blurs caused by camera motion during exposure time and restores a latent sharp image from a noisy blurry one. Typically, a blurry image is modeled as a latent sharp image convolved with a blur kernel plus noise. In most cases, the blur kernel is unknown. Thus image deblurring is an ill-posed problem since both the sharp image and the blur kernel have to be estimated from a single blurry observation.

To make this inverse problem well posed, researchers investigated the possibility to extract auxiliary information from extra images or camera built-in inertial sensors. The extra images typically refer to additional noisy or blurry frames with different exposure settings except for the target blurry image. The work of Yuan *et al.* [2] is one of the first that employed noisy/blurry image pairs to estimate the blur kernel. Zhang *et al.* [3] proposed a Bayesian deblurring algorithm that utilized a flexible number of degraded noisy or blurry frames to restore the latent sharp one. As inertial sensors can provide clues of camera motion, its usage is also explored for kernel estimation. The early work [4] attached an extra inertial measurement sensor unit, including gyroscope and accelerometer, to a digital camera. Today, almost all of smartphones are equipped with both the camera and the inertial sensors. Thus, instead of using extra bulky sensors, Sindelar *et al.* [5] simplified the algorithm in [4] and applied it to smartphone devices. More recently, Zhen *et al.* [1] proposed a blind image deblurring scheme that benefits from both extra noisy images and smartphone built-in inertial sensors to jointly solve depth estimation and image deblurring problem.

Recent progress of deep learning application in image deblurring has drawn great attention. The pioneering work [6] and

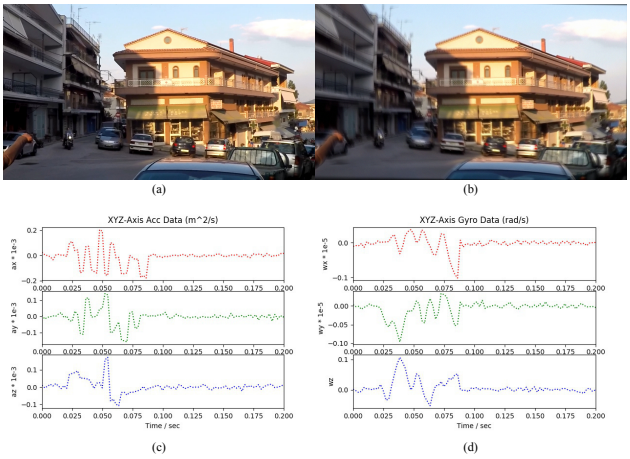


Figure 1. *The Proposed Data. (a) Groundtruth sharp image. (b) Blurry image. (c) Gyroscope Data. (d) Accelerometer Data.*

[7] trained neural networks to predict a blur kernel from an input blurry image and then estimated the latent sharp image using the conventional deconvolution process. To avoid the time-consuming deconvolution, the end-to-end or image-to-image networks are more widely used today. Nah *et al.* [8] proposed a multi-scale end-to-end network to mimic conventional coarse-to-fine deblurring scheme. Tao *et al.* [10] improved its performance by an embedded Long-Short Term Memory (LSTM) unit [11]. Inspired by the development of General Adversarial Nets (GAN) [12], Kupyn *et al.* [9] proposed a GAN based deblurring network called DeblurGAN. In order to suppress the artifacts in the result of DeblurGAN, Zhang *et al.* [13] incorporated the dark channel prior [14] into the network through the lost function. Multiple input images were also considered in the deep learning network. A hybrid network of LSTM and GAN was proposed by Zhang *et al.* [15] to boost the deblurring performance by extracting useful information from a noisy/blurry image pair.

The inertial sensor data aided deblurring scheme has achieved less success in deep learning field. As far as we know, Mustaniemi *et al.* [16] is the only one who proposed a deep deblurring network which takes gyroscope data as well as a blurry image as input. The limitation of their work is that it can only handle motion blurs caused by camera rotation. Therefore, the potential of inertial sensor data is not fully demonstrated. One reason is that applying inertial sensor data to deblurring network is challenging since convolution neural networks trend to process one-dimensional inertial sensor data or three-dimensional color images but not both at the same time, let alone the problem of synchronization, noise and other error effects between or within the

[1] https://drive.google.com/file/d/18_PcNpadgxPOSaSpsUcFiTHpxNDmMtO3/view?usp=sharing

two types of data. Another reason is a lack of datasets. The training dataset is essential for deep learning networks. Even though the benchmark dataset GOPRO has been proposed [8] and widely used [8, 9, 10], it's mainly designed for networks whose input are only images.

To fill in this blank, in this paper, we propose a synthetic training and testing dataset for deep deblurring networks aided by camera built-in inertial sensors. The proposed dataset contains gyroscope and accelerometer data as well as sharp/blurry image pairs, in which the realistic error effects, including noise, misalignment, rotation center shift and rolling shutter effect, are simulated.

2. Idea Model

This section discusses the geometric blur model which has been widely used for inertial sensor aided image deblurring algorithms [1][18] and the model for generating the synthetic gyroscope and accelerometer data. Adding the effect of common error will be discussed in the next section.

2.1 Geometric Blur Model

Recall the convolutional model of the blurry image: $\boldsymbol{I_B} = \boldsymbol{I_S} * \boldsymbol{k} + \vec{\boldsymbol{n}}$, where $\boldsymbol{I_B}$ and $\boldsymbol{I_S}$ denote the blurry image and its latent sharp image, respectively. $\boldsymbol{k}$ is the blur kernel which parameterizes the blur model. The symbol $*$ represents the convolution operator. And the noise $\vec{\boldsymbol{n}}$ is often formulated as the additional white Gaussian noise.

To simulate a more realistic blurry image using inertial sensors, instead of the conventional convolutional model, a geometric blur model is adopted. The geometric blur model formulates the blurry image as the integration of the sharp image under a sequence of projective motions during the exposure time:

$$\boldsymbol{I_B} = \sum_{i=1}^{N_p} w_i \boldsymbol{I_S}(\boldsymbol{H}_i \boldsymbol{x}) + \vec{\boldsymbol{n}}, \tag{1}$$

where the 3×1 vector $\boldsymbol{x}$ denotes the homogeneous pixel coordinate. $\boldsymbol{I_S}(\boldsymbol{H}_i \boldsymbol{x})$ can be viewed as the intermediate transformed frame captured by the camera at one pose which is characterized by the 3×3 homography matrix $\boldsymbol{H}_i$ at the pose i and the corresponding weight w_i which is proportional to exposure time at that pose. N_p denotes the number of all camera poses during the exposure time. The first term of the convolutional model is replaced by estimation of homography matrices $\{\boldsymbol{H}_i\}$ and their corresponding weights $\{w_i\}$. And the homography matrix $\boldsymbol{H_i}$ can be characterized by the rotation matrix $\boldsymbol{R}_i$, the translation vector $\vec{\boldsymbol{t}}_i$, the normal vector $\vec{\boldsymbol{n}}_{\boldsymbol{t}}$, the scene depth d and a intrinsic matrix $\boldsymbol{\Pi}$ [17]:

$$\boldsymbol{H_i} = \boldsymbol{\Pi}(\boldsymbol{R}_i + \frac{\vec{\boldsymbol{t}}_i \, \vec{\boldsymbol{n}}_t^{\,T}}{d})\boldsymbol{\Pi}^{-1}, \tag{2}$$

where the intrinsic matrix $\boldsymbol{\Pi}$ can be represented by the focal length f and camera optical center (o_x, o_y). The depth d is constant which can be predicted by the smartphone camera. To simplify the model, the depth is set to 1 and the normal vector is always vertical to the image plane. Typically, camera rotations $\boldsymbol{R}_i$ and translations $\vec{\boldsymbol{t}}_i$ can be estimated from measurement of gyroscopes and accelerometers, respectively. The gyroscope measures rotation rates around x, y, z axis. And the accelerometer records both the physical acceleration along the three a axis and the contributor of gravitational force. Some mobile platforms, like Andriod, have provided a software based linear accelerometer that eliminates the gravity using other sensor's data. In this paper, the sample data from the linear accelerometer is chosen for simplicity. The measured 3-axis angular velocity and the linear acceleration at the pose index i are denoted as $\vec{\boldsymbol{\omega}}_i = [\omega_{ix}, \omega_{iy}, \omega_{iz}]$ and $\vec{\boldsymbol{a}}_i = [a_{ix}, a_{iy}, a_{iz}]$.

Given the sampling interval Δt and the rotation matrix $\boldsymbol{R}_i$ at index i, the rotation matrix at next index $i+1$ can be approximated as [5]:

$$\boldsymbol{R}_{i+1} = \boldsymbol{R}_i + \Delta t * \frac{d\boldsymbol{R}_t}{dt} = \begin{bmatrix} 1 & -d\phi_{iz} & d\phi_{iy} \\ d\phi_{iz} & 1 & -d\phi_{ix} \\ -d\phi_{iy} & d\phi_{ix} & 1 \end{bmatrix} \boldsymbol{R}_i, \tag{3}$$

where $d\vec{\boldsymbol{\phi}}_i = [d\phi_{ix}, d\phi_{iy}, d\phi_{iz}] = \vec{\boldsymbol{\omega}}_i * \Delta t$.

Once the rotation matrix $\boldsymbol{R}_i$ is known, the camera translation $\vec{\boldsymbol{t}}_i$ can be derived from accelerations $\vec{\boldsymbol{a}}_i$ by twice integration:

$$\begin{aligned} \vec{\boldsymbol{v}}_i &= \vec{\boldsymbol{v}}_{i-1} + (\boldsymbol{R}_{i-1}^{-1}\vec{\boldsymbol{a}}_{i-1} + \boldsymbol{R}_i^{-1}\vec{\boldsymbol{a}}_i) * \Delta t/2 \\ \vec{\boldsymbol{t}}_i &= \vec{\boldsymbol{t}}_{i-1} + (\vec{\boldsymbol{v}}_{i-1} + \vec{\boldsymbol{v}}_i) * \Delta t/2 \end{aligned} \tag{4}$$

where the initial velocity $\vec{\boldsymbol{v}}_0$ and initial translation $\vec{\boldsymbol{t}}_0$ is assumed to be 0.

2.2. Synthetic Gyroscope and Accelerometer Data

In the proposed dataset, the blurry image and corresponding inertial sensor data are generated from synthetic samples of gyroscopes and accelerometers. The angular velocity and the acceleration of each axis are modeled by Gaussian distribution with zero mean. The standard deviation of angular velocity of each axis is $\sigma_{\omega_x} = \sigma_{\omega_y} = 0.05$e-5 and $\sigma_{\omega_z} = 0.05$ rad/s ; and the standard deviation of acceleration of each axis is $\sigma_{a_x} = \sigma_{a_y} = \sigma_{a_z} =$ 1e-4 m^2/s. The angular velocity and the acceleration are sampled randomly within the exposure time, where the exposure time t_e is picked randomly from the range $[0.01, 0.1]$ second and the sampling frequency $f_s = 200$ Hz. Typically, 20 to 40 samples can be collected from the smartphone inertial sensors. The number of poses picked in this dataset is 30. To attain enough samples and mimic the continuous movement, the sample represented each pose is interpolated from previous data points, linearly for angular velocity data and nearly for acceleration data.

3. Error Effects

The previous section presents an ideal blurry image model that the synthetic inertial sensor data can be incorporated into the geometric blur model directly. This model, however, implicitly assumes that the inertial sensor readings are well aligned with the camera motions and are not degraded by any noise, that the camera rotation center locates at the the optical center, and that the image sensor employs global shutter to capture entire frame all at once. To simulate more realistic situations, this section discusses the practical error effects involved in using inertial sensors to images captured by an smartphone camera.

3.1. Time Delay

Due to the different launch time, the inertial sensor data is not always well aligned with smartphone movements. The inertial

sensor usually lags behind the image senor. This misalignment is typically modeled as a constant time delay t_d between inertial and image sensor measurements [19]. The value of t_d varies with different smartphones or temperature, but it's usually in the scale of 10^{-2} second [18][19]. In the proposed dataset, the time delay t_d is randomly picked from a Gaussian distribution $\mathcal{N}(0.03, 0.01^2)$ in second. In the previous section, the ideal inertial sensor data contains $N_p + 1$ samples to generate N_p camera poses. To add time delay to the data, the total samples is extended from 31 to 220 by padding zeros and the ideal inertial sensor data is shifted right from 0 to the timestamp represents the length of time delay.

3.2. Rotation Center Shift

Although the camera optical center is often assumed as the rotation center like [4], it's not always true in real scenarios. For example, the rotation center could locate at the wrist of people who is holding the smartphone. The inertial-aided image deblurring algorithms, like Park *et al.* [20] and Hu *et al.* [18], adopted a more accurate way that assumes the center of rotation locates at the image plane and is fixed during the exposure time. In the proposed dataset, the shift of the rotation center $(\Delta o_x, \Delta o_y)$ is randomly picked from a Gaussian distribution whose mean is the center of image and the standard deviation is $0.25\times$image width and $0.25\times$image height in pixel. A new intrinsic matrix Π^* is calculated from the shifted rotation center $(o_x - \Delta o_x, o_y - \Delta o_y)$.

3.3. Rolling Shutter Effect

Smartphones usually are equipped with low-cost cameras which utilize the rolling shutter mechanism that each image row is exposed at a slightly different time. The proposed dataset also takes this error effect into account since the rolling shutter mainly effects the image captured by a moving camera during the exposure. The rolling shutter effect can be characterized by the camera readout time t_{read}. For a point $\boldsymbol{x}_i = (u_i, v_i, 1)^{\mathrm{T}}$ in the frame of the timestamp i, the time at which point $\boldsymbol{x}_i$ was imaged depends on the timestamp of frame t_i and its row index v:

$$t(i, v_i) = t_i + t_{read} \cdot \frac{v_i}{h}, \tag{5}$$

where h is the image height. If $\boldsymbol{x}_i$ is the projection of a real point $\boldsymbol{X}$ and there exists another point $\boldsymbol{x}_j$ on the frame j which is also the projection of $\boldsymbol{X}$, the relation between the two points can be written as:

$$\boldsymbol{x}_j = \boldsymbol{W}(t(j, v_j), t(i, v_i))\boldsymbol{x}_i \tag{6}$$

$$\boldsymbol{W}(t(j, v_j), t(i, v_i)) = \boldsymbol{H}(t(j, v_j))\boldsymbol{H}(t(i, v_i))^{-1}, \tag{7}$$

where $\boldsymbol{W}(\cdot)$ and $\boldsymbol{H}(t)$ denote the wrapping matrix from the frame i to the frame j and the homography matrix at time t, respectively. Assuming the frame i is taken by a global shutter camera, like in the previous section, and the frame j is projection of the same scene as the frame i but captured a rolling shutter camera, the wrapping matrix becomes $\boldsymbol{W}(t(i, v_i), t_i)$ [21].

This new wrapping matrix can map the idea blurry image generated in the previous section to a new one that suffers from the rolling shutter effect. The only parameter missing here is the readout time t_{read}. In the proposed dataset, the readout time is randomly picked from a Gaussian distribution $\mathcal{N}(0.015, 0.006^2)$.

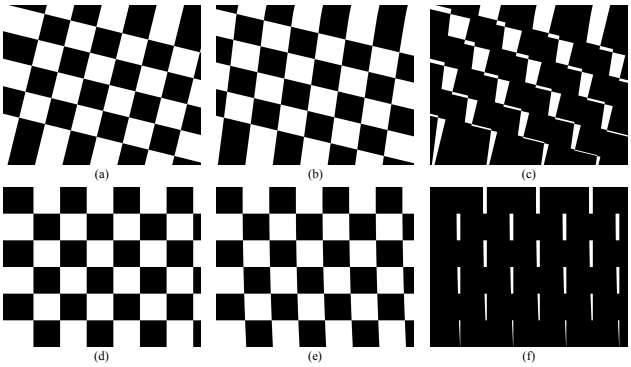


Figure 2. *The rolling shutter effect. (a) Rotated image. (b) Add rolling shutter effect in (a). (c) Difference between (a) and (b). (d) Translated image. (e) Add rolling shutter effect in (d). (f) Difference between (d) and (d). Top to bottom: pure rotation around z axis and pure translation along x axis.*

Figure 2 shows the difference between with and without considering the rolling shutter effect for the pure rotation (top) and the pure translation (bottom). Most image deblurring algorithms only consider pure rotation of the camera to simplify the model [16][21]. However, Figure 2 shows that translations of camera also matters. That's the reason why the rotation matrix in [21] is replaced with the homography matrix to apply rolling shutter effect in Equation (6) and (7).

3.4. Noise

Both the inertial sensor data and the blurry image are degraded by noise in practical situations. The effect of noise for accelerometer data is more severe since it requires twice integration to compute translation vector from the acceleration. The noise for inertial sensor data is model as additive white Gaussian noise with standard deviation as $1/10$ of the standard deviation of corresponding data which are specified in the previous section [22]. And the noise for the blurry image is formulated as additive white Gaussian noise with standard deviation σ_r/N_p, where σ_r is uniformly sampled from $[0.05, 0.1]$ and N_p is the number of poses [13].

4. Dataset Details

In the proposed dataset, the groundtruth sharp frames used to generate blurry ones are picked from the GOPRO dataset [8]. To mimic the image captured by smartphone camera, the focal length is set to 50 mm and the pixel size is 2.44e-6 m/pixel. The proposed dataset contains 2264 sets of data in the train dataset and 1221 sets in the test dataset. Each set of data includes a sharp frame, an intermediate blurry image without errors, a blurry image with error effects, intermediate inertial sensor data without error effects and inertial sensor data with error effects. The resolution of the images is 720x1280.

Figure 3 demonstrates the flow chart of the proposed dataset. An intermediate blurry image is generated from groundtruth sharp frame and synthetic original inertial sensor data through the geometric blur model. After that, the error effects are added to this original blurry image in the order of shift rotation center, rolling shutter effect and noise. The order of the first two effects cannot change since the rolling shutter requires the new intrinsic matrix

calculated from the shifted rotation center. The time delay and noise are applied to the original inertial sensor data to generate the final inertial sensor data.

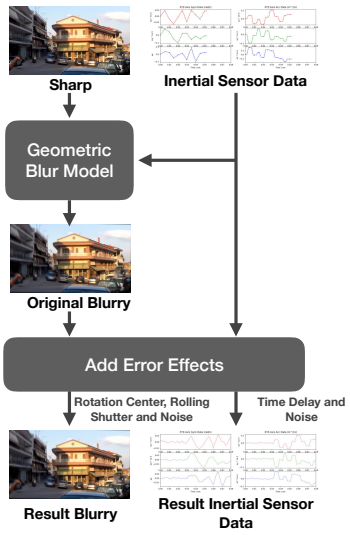


Figure 3. *The proposed flow chart.*

5. Result

Examples of the proposed datasets are presented in Figure 4, where the first and the second columns are extreme cases to verify the correctness of the data and the error effects, and the last column is a more general case. Rows from top to bottom denote the groundtruth sharp frame, the original blurry frame without error effects, the result blurry frame considering error effects, the gyroscope data and the accelerometer data.

Figure 4(a) shows an example of a pure camera rotation around z axis. This is an extreme case since the angular rate in z axis is manually set to be a comparably high value during the exposure time and the rest angular rate as well as all the acceleration are set to zeros. The second row of Figure 4(a) presents a blur artifact caused by a pure rotation around the center of image, where $(o_x, o_y) = (0.5 \cdot \text{Image Width}, 0.5 \cdot \text{Image Height}) = (640, 360)$. After the new rotation center shifted $(\Delta o_x, \Delta o_y) = (-278, -185)$, the center of the blur caused by rotation is translated upper and left correspondingly in the third frame. Compared to the second frame, the upper left angle of third demonstrates less black edge, which results from both the shifted rotation center and the rolling shutter effect.

Figure 4(b) is an example of a pure translation along x axis, where the acceleration of x axis is set to be a high value and the rest inertial sensor data is set to zeros. The blurry images, the second and the third frame, show a blur artifact caused by translation. Compared to the second frame, the left side of the third frame presents less dark area.

Figure 4(c) demonstrates a more general case in the proposed dataset. Noticeably, the inertial sensor data added time delay (denoted by the solid line) lags behind of the one without error effects (denoted by the dashed line). The original inertial sensor data presented by the dashed line also ends much earlier. That's because the number of poses during the exposure is set to 30 in the proposed dataset and it is padded to 220 samples to add the timestamp shift caused by time delay.

6. Conclusion

In this paper, a synthetic image deblurring dataset aided by inertial sensors is proposed in order to pave the way for development of the camera built-in inertial-aided deblurring scheme in deep learning field. To generate the blurry image, a groundtruth sharp frame combined with the synthetic gyroscope and accelerometer data are fed into the geometric blur model. To mimic more realistic data, error effects like misalignment, rotation center shift and rolling shutter effect are taken into account in the proposed dataset. Both the data with or without error effects are collected in the proposed dataset to help the user check the intermediate result. The proposed dataset is verified by two extreme cases and a general case.

References

[1] Ruiwen Zhen and Robert Stevenson, Inertial Sensor Aided Multi−image Nonuniform Motion Blur Removal Based on Motion Decomposition, Journal of Electronic Imaging, 27, pg.053026. (2018).

[2] Lu Yuan, Jian Sun, Long Quan, and Heung-Yeung Shum, Image Deblurring with Blurred/Noisy Image Pairs, ACM Trans.Graph., 26 (2007).

[3] Haichao Zhang, David Wipf, Yanning Zhang, Multi-image Blind Deblurring Using a Coupled Adaptive Sparse Prior, In 2013 CVPR, pg. 10511058. (2013).

[4] Neel Joshi, Sing Bing Kang, C. Lawrence Zitnick, andRichard Szeliski, Image Deblurring Using Inertial Measurement Sensors, ACM Trans. Graph., 29, pg. 30:130:9. (2010).

[5] Ondrej Sindelar and Filip Sroubek, Image Deblurring in Smartphone Devices Using Built-in Inertial Measurement Sensors, J. Electronic Imaging, 22, pg. 011003. (2013).

[6] Jian Sun, Wenfei Cao, Zongben Xu, and Jean Ponce, Learning a Convolutional Neural Network for Non-uniform Motion Blur Removal, In 2015 CVPR, pg. 769777. (2015).

[7] Christian J. Schuler, Michael Hirsch, Stefan Harmeling, Bernhard Schlkopf, Learning to Deblur, IEEE Trans. on Pattern Analysisand Machine Intelligence, 38, pg. 14391451. (2016).

[8] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee, Deep Multi-scale Convolutional Neural Network for Dynamic Scene Deblurring, In 2017 CVPR, pg. 257265. (2017).

[9] Orest Kupyn, Volodymyr Budzan, Mykola Mykhailych, and et al., DeblurGAN: Blind Motion Deblurring Using Conditional Adversarial Networks, In 2018 CVPR, pg. 81838192. (2018).

[10] Xin Tao, Hongyun Gao, Xiaoyong Shen, Jue Wang, and Jiaya Jia, Scale-Recurrent Network for Deep Image Deblurring, In 2018 CVPR, pg. 81748182. (2018).

[11] Sepp Hochreiter and Jurgen Schmidhuber, Long Short-Term Memory, Neural Comput., 9, pg. 17351780. (1997).

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, and et al., Generative Adversarial Nets, Advances in Neural Information Processing Systems 27, Curran Associates, Inc., 2014, pg. 26722680.

[13] Shuang Zhang, Ada Zhen, and Robert L. Stevenson, GAN Based Image Deblurring Using Dark Channel Prior, Fast Track Article for IS&T International Symposium on Electronic Imaging 2019: Computational Imaging XVII Proceedings., pg. 13611365. (2019).

[14] Jinshan Pan, Deqing Sun, Hanspeter Pfister, et al., Blind Image De-

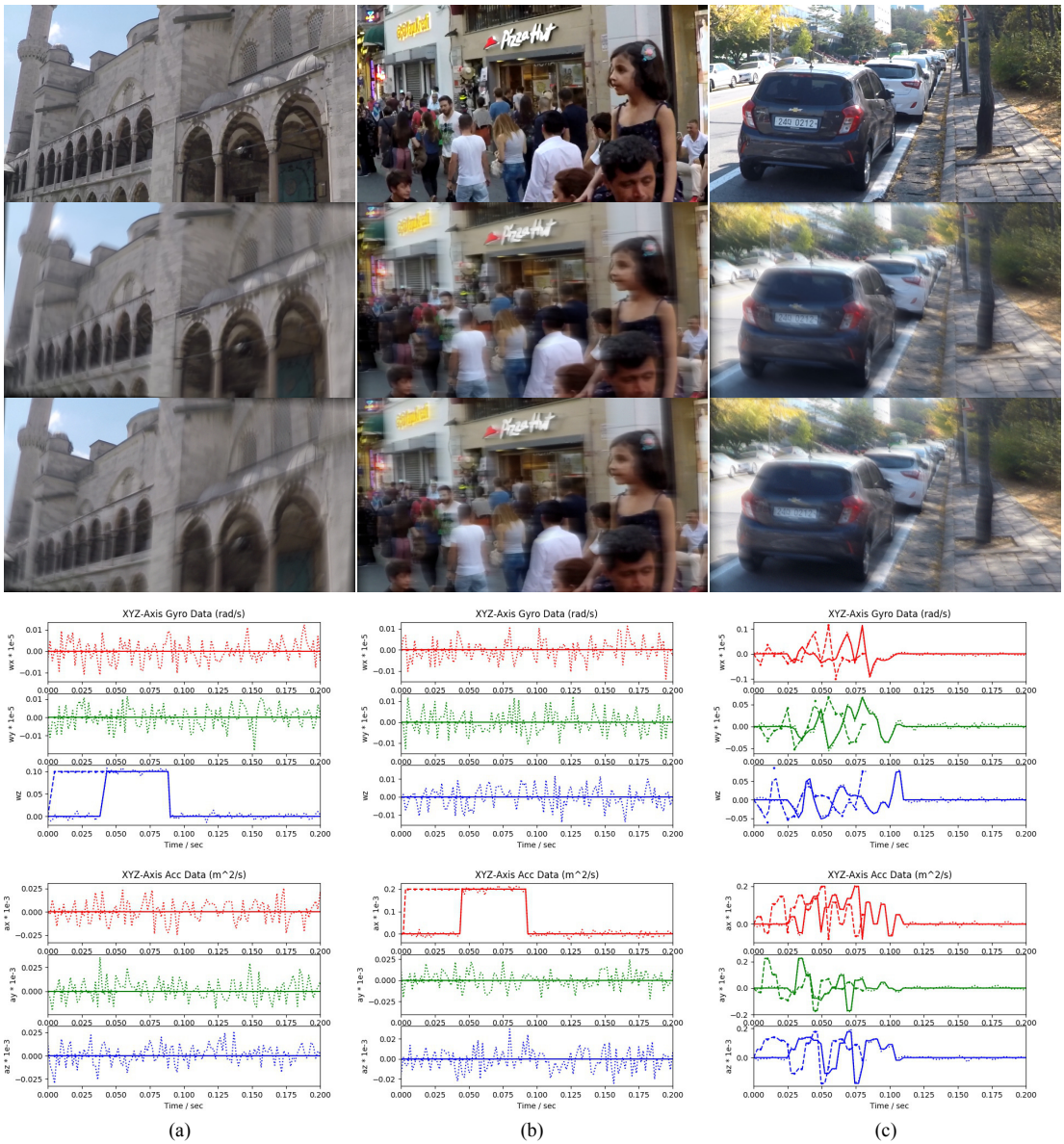


Figure 4. *Examples taken from the proposed dataset. (a) Pure rotation around z axis. (b) Pure translation along x axis. (c) General case. Top to Bottom: groundtruth sharp frame, original blurry image, blurry image with error effects, gyroscope data and accelerometer data, where dashed line, solid line and dotted line represent original inertial sensor data, data added time delay and data considering both time delay and noise, respectively.*

blurring Using Dark Channel Prior, In CVPR 2016, pg. 1628−1636. (2016).

[15] Shuang Zhang, Ada Zhen, and Robert L. Stevenson, DeepMotion Blur Removal Using Noisy/Blurry Image Pairs, ArXiv: https://arxiv.org/abs/1911.08541. (2019).

[16] Janne Mustaniemi, Juho Kannala, Simo Srkk, and et al., Inertial-aided Motion Deblurring with Deep Networks, CoRR, abs/1810.00986. (2018).

[17] Olivier D. Faugeras and Francis Lustman, Motion and Structure from Motion in a Piecewise Planar Environment, IJPRAI, 2, pg. 485508. (1988).

[18] Zhe Hu, Lu Yuan, Stephen Lin, and Ming-Hsuan Yang, Image Deblurring Using Smartphone Inertial Sensors, In 2016 CVPR, pg. 18551864. (2016).

[19] Jonathan Kelly, Nicholas Roy, and Gaurav S. Sukhatme, Determining the TimeDelay Between Inertial and Visual Sensor Measurements,IEEE Trans. on Robotics, 30, pg. 15141523. (2014).

[20] Sung Hee Park and Marc Levoy, Gyro-Based Multi-image Deconvolution for Removing Handshake Blur, In CVPR 2014, pg. 33663373. (2014) .

[21] Alexandre Karpenko and David Jacobs, Digital Video Stabilization and Rolling Shutter Correction Using Gyroscopes, Stanford University CTSR 2011-03, pg. 1-7. (2011).

[22] Manon Kok, Jeroen D. Hol, and Thomas B. Schön, Using Inertial Sensors for Position and Orientation Estimation, Found. Trends Signal Process., 11, pg. 1153. (2017).

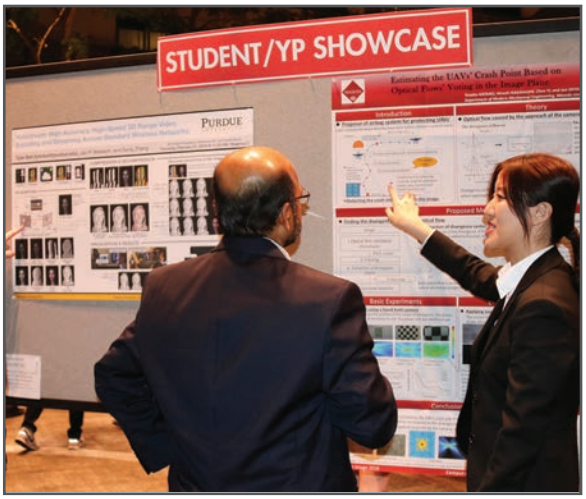
JOIN US AT THE NEXT EI!

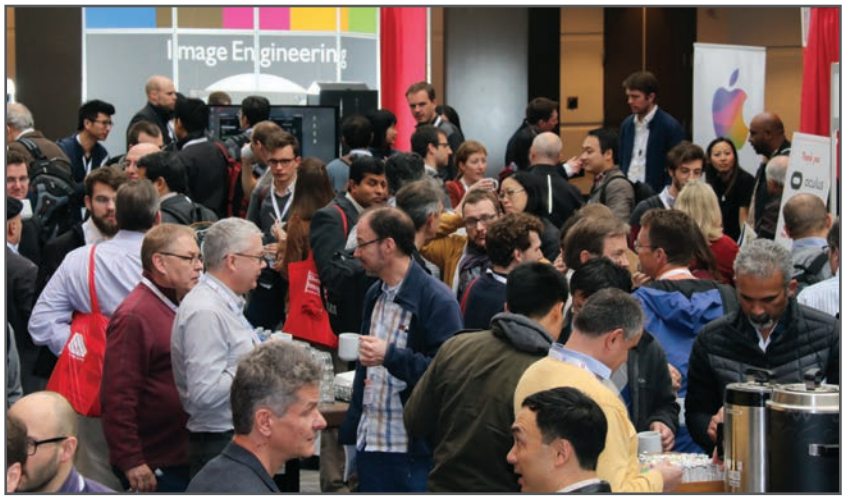
IS&T International Symposium on

Electronic Imaging

SCIENCE AND TECHNOLOGY

Imaging across applications . . . Where industry and academia meet!





• SHORT COURSES • EXHIBITS • DEMONSTRATION SESSION • PLENARY TALKS •
• INTERACTIVE PAPER SESSION • SPECIAL EVENTS • TECHNICAL SESSIONS •

www.electronicimaging.org

IS&T
imaging.org