

# Gyroscope-Aided Motion Deblurring with Deep Networks

<sup>1</sup>Janne Mustaniemi   <sup>2</sup>Juho Kannala   <sup>2</sup>Simo Särkkä   <sup>3</sup>Jiri Matas   <sup>1</sup>Janne Heikkilä

<sup>1</sup>Center for Machine Vision and Signal Analysis, University of Oulu, Finland   <sup>2</sup>Aalto University, Finland

<sup>3</sup>Center for Machine Perception, Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic

janne.mustaniemi@oulu.fi

## Abstract

*We propose a deblurring method that incorporates gyroscope measurements into a convolutional neural network (CNN). With the help of such measurements, it can handle extremely strong and spatially-variant motion blur. At the same time, the image data is used to overcome the limitations of gyro-based blur estimation. To train our network, we also introduce a novel way of generating realistic training data using the gyroscope. The evaluation shows a clear improvement in visual quality over the state-of-the-art while achieving real-time performance. Furthermore, the method is shown to improve the performance of existing feature detectors and descriptors against the motion blur.*

## 1. Introduction

Motion blur is often unavoidable when capturing images with a fast moving camera. It not only degrades the visual quality but it also has a negative impact on applications such as visual odometry, augmented reality (AR) and simultaneous localization and mapping (SLAM). Even though the blind deblurring methods have improved significantly over the years, they generally struggle with strong and spatially-variant motion blur. We intend to overcome these limitations by utilizing inertial measurements.

Blind deconvolution methods aim to recover the sharp image without any additional information about the motion blur. This is an ill-posed problem since the blurred image only provides a partial constraint on the solution. Promising results have been obtained with recent deep learning based approaches [12, 15]. These methods are especially good at generating perceptually convincing images while avoiding deblurring artifacts. To simplify the problem, the existing methods typically assume a spatially-invariant blur, which may not hold in practice. An example of such case is shown in Figure 1.

Mobile devices are often equipped with an inertial measurement unit (IMU), which provides information about the motion blur. Accelerometers and gyroscopes have been suc-



Figure 1. Heavily blurred images captured with a fast moving camera (top). Images deblurred by DeepGyro CNN (bottom).

cessfully used in motion deblurring [10, 20, 6, 7, 31, 14]. Most of these methods focus on the removal of the camera shake blur. An application such as SLAM may involve a fast moving camera, which generally results in much stronger motion blur. The existing methods are also not capable of running in real-time, apart from [14]. What further complicates the problem is that inertial-based blur estimates may be inaccurate. This can be due to noisy IMU readings, temporal misalignment between the camera and IMU, unknown scene depth or translation. These limitations should be considered in order to avoid deblurring artifacts.

We propose a deblurring method that incorporates gyroscope measurements into a convolutional neural network (CNN). It can handle extremely strong and spatially-variant motion blur as illustrated in Figure 1. When computing the gyro-based blur estimates, we take into account that mobile devices are usually equipped with a rolling shutter camera. The method naturally overcomes the limitations of gyro-based blur estimation by utilizing image data. We also introduce a novel data generation scheme, which is an essential component needed to train our network. The evaluation on real-world images shows a clear improvement in visual quality over the state-of-the-art while achieving real-time performance. The method will also improve the robustness of existing feature detectors and descriptors against motion blur as indicated by the higher repeatability and better matching performance.

## 2. Related work

Despite being a classical image processing problem, deblurring continues to be an active research area with plenty of recent progress. For example, regarding blind single-image deblurring, the recent papers utilizing so called dark and bright channel priors have shown promising results [17, 29]. Nevertheless, these approaches typically assume uniform and spatially invariant blur, which is often not the case in practice. For example, if there is rotation around optical axis, the blur kernel is clearly spatially variant.

Recently, there have emerged also several deep learning based blind deblurring methods. For example, the concept of generative adversarial networks has been utilized for learning deep neural networks that perform deblurring [12, 16, 15]. In particular, inspired by *pix2pix* [9], *DeblurGAN* [12] trains a conditional GAN for deblurring using pairs of corresponding blurred and sharp images. However, as the blind deblurring problem is severely ill-posed, the results are often not good or satisfactory. In fact, we use *DeblurGAN* as one of the baseline methods and Figures 5 and 8 illustrate that its results are clearly inferior to ours.

Besides methods that directly perform blind deblurring, there are also approaches that first estimate a spatially-variant motion field and blur kernels from a single image using deep networks, and thereafter perform non-blind deconvolution [24, 4]. Further, deep nets have been trained to remove deblurring artifacts that non-blind deconvolution typically creates, either directly predicting the sharp output image [22] or the residual image between the deconvolution result and the desired sharp output [28].

In addition to single-image deblurring methods, there are also methods that utilize additional information, such as multiple frames from a video [2, 23], bursts of rapidly captured photographs [1], pairs of blurred and noisy images captured with different exposure settings [30], or high- and low-resolution image pairs [25]. While some of the aforementioned methods provide promising results, they belong to a different domain than our single-image deblurring approach. Moreover, multiple images are not always available or easy to capture as dynamic objects and events may disappear from the scene.

Our work deals with inertial-aided single-image deblurring. That is, we learn a deep neural net for deblurring a single RGB image so that the input to the net is the blurred image and a spatially varying motion field estimated based on gyroscope measurements recorded during the exposure of the image. This problem setting is highly relevant in practice since rotation is usually the main source of blur due to hand shake [6] and most smartphones are equipped with gyroscopes. There have been relatively many papers that utilize inertial sensors (gyroscopes and/or accelerometers) for image deblurring [10, 7, 6, 31, 14, 20]. Most of them focus on estimating and characterizing the blur ker-

nels based on the inertial sensor data [10, 7, 6, 14, 20] and then apply non-blind deconvolution. Nevertheless, due to the limitations of consumer grade inertial sensors in smartphones, the motion estimates can never be perfect and, in practice, there may also be dynamic objects in the scene and their apparent motion is not explained by device motion. Thus, it seems plausible to combine inertial measurements and image based information for deblurring [31] and our work does that by utilizing deep CNNs. To the best of our knowledge, our method is the first one that combines gyroscope measurements and learnt neural network based image priors for deblurring. This approach has significant benefits as our results show a clear improvement in visual quality over the previous state-of-the-art while achieving real-time performance.

## 3. Blur estimation

Motion blur is caused by the relative motion of the camera and scene during the exposure of the image. This work focuses on static scenes, meaning the motion blur is only due to the rotation and translation of the camera. The initial estimate for the motion blur is obtained with the gyroscope. A key challenge is to represent this information in a format useful for the deep network. This process will be covered in the next section. As a result, we get a spatially-variant blur field, which is provided for the deblurring network as an additional input.

### 3.1. Rotation from gyroscope measurements

In prior work [20, 6], it has been shown that motion blur is typically caused by the rotation of the camera. Similar to these works, we compute the rotations by integrating gyroscope readings. More specifically, we numerically integrate the quaternion differential equation (e.g. [26])

$$\frac{d\mathbf{q}(t)}{dt} = \frac{1}{2} \mathbf{q}(t) \odot \boldsymbol{\omega}(t), \quad \mathbf{q}(t_1) = \mathbf{1}, \quad (1)$$

where  $\boldsymbol{\omega}(t)$  is the 3-dimensional gyroscope measurement and  $\odot$  denotes the quaternion product. The initial condition is given at the starting time of exposure  $t_1$  and the solution is computed at the end time of exposure  $t_2$ . The rotation matrix  $\mathbf{R}(t_2)$  is then formed as the direction cosine matrix corresponding to the quaternion  $\mathbf{q}(t_2)$  (see, e.g., [26]).

In theory, the translation could also be recovered using an accelerometer [31, 7, 10, 21]. However, this requires knowledge of the initial velocity of the camera, or alternatively, known stationary points or reference points which can be used to aid zero-velocity updates or position updates in a Kalman filter [21]. However, these are not assumed to be available here. Furthermore, the motion blur caused by translation will also depend on the scene depth, which is difficult to estimate from a single image. We take these limitations into account when generating training data.

### 3.2. Blur field computation

If the camera is moving during the image exposure, the 3D scene points will be projected to multiple points on the image plane. This will appear as motion blur. To estimate the blur, we need to consider the relative motion of the camera during the exposure. Let  $\mathbf{R}(t)$  and  $\mathbf{t}(t)$  denote the rotation and translation of the camera. Assuming that the scene has a constant depth  $d$ , the motion can be modeled using a planar homography [5]

$$\mathbf{H}(t) = \mathbf{K}[\mathbf{R}(t) - \frac{\mathbf{t}(t)\mathbf{n}^\top}{d}]\mathbf{K}^{-1}, \quad (2)$$

where  $\mathbf{K}$  is the intrinsic camera matrix obtained via calibration. The normal vector of the scene is denoted by  $\mathbf{n}$ . If the translation is zero (or if the scene is far away), the previous equation simplifies to

$$\mathbf{H}(t) = \mathbf{K}\mathbf{R}(t)\mathbf{K}^{-1} \quad (3)$$

Let  $\mathbf{x} = (x, y, 1)^\top$  be the projection of the 3D point at the beginning of the exposure. The rest of the projections can be computed by  $\mathbf{x}' = \mathbf{H}(t)\mathbf{x}$ .

If the exposure time is relatively short (e.g. when capturing a video), the motion blur can be assumed to be linear and homogeneous. This type of blur can be described with a 2-dimensional blur vector  $(u, v)$ , where  $u$  and  $v$  represent the horizontal and vertical components of the blur, respectively. See the visualization in Figure 2. Note that all blur vectors with equal lengths and opposite directions, such as  $(u, v)$  and  $(-u, -v)$  will correspond to the same blur. Therefore, we choose to constrain the horizontal component  $u$  to be positive. We compute the blur vectors for every pixel, which gives us the blur maps  $\mathbf{U}$  and  $\mathbf{V}$  in horizontal and vertical directions. Together these are referred to as blur field  $\mathcal{B} = (\mathbf{U}, \mathbf{V})$ .

### 3.3. Rolling shutter effect

Mobile devices are typically equipped with a rolling shutter camera. This means, each row of pixels will be captured at slightly different time. The formula 3 cannot be used directly since the mapping of the point  $\mathbf{x}$  depends on its  $y$ -coordinate. Let  $t_r$  denote the camera readout time, that is the time difference between the exposure of the first and last row of pixels. Then, the exposure of the  $y$ :th row starts at

$$t_1(y) = t_f + t_r \frac{y}{N}, \quad (4)$$

where  $t_f$  is the frame timestamp and  $N$  is the number of rows. The end of the exposure is defined as  $t_2 = t_1 + t_e$ , where  $t_e$  is the exposure time. The mapping of the point then becomes

$$\mathbf{x}' = \mathbf{K}\mathbf{R}(t_2)\mathbf{R}^\top(t_1)\mathbf{K}^{-1}\mathbf{x}. \quad (5)$$



Figure 2. Comparison of the real blur and gyro-based blur kernel estimates  $(u, v)$ , in red. The local real blur is best visible at light-streaks.

Note that the frame timestamp  $t_f$ , readout time  $t_r$  and exposure time  $t_e$  can be typically obtained via the API of the mobile device.

## 4. Deblurring

Deblurring is based on a fully-convolutional neural network. It aims to produce a sharp image given a blurred image and gyro-based blur field. The architecture of the network is described in the next section. To train the network, we propose a data generation scheme that utilizes gyroscope readings. This topic is covered in Section 4.2.

### 4.1. Network architecture

The architecture of the network is shown in Figure 3. The network is similar to U-Net [18], which was originally used for image segmentation. This type of encoder-decoder network has proven to be useful in various image-to-image translation problems [9]. The input of our network consists of a blurred RGB image and a gyro-based blur field. They pass through a series of convolutional and downsampling layers, until the lowest resolution is reached. After the bottleneck, this process is reversed. A low-resolution image is expanded back into a full resolution image with help of upsampling layers. Skip connections are used to allow information sharing between the encoder and decoder. Given two layers with equal size, the feature maps from the encoder are concatenated with those of the decoder. The input images can be of arbitrary size since the network is fully-convolutional.

### 4.2. Data generation

To train the network, we need a set of blurred and sharp images along with gyro-based blur fields. There is no easy way to capture such real-world data. As mentioned, the motion blur is mainly caused by the rotation of the camera. We utilized gyroscope readings to generate realistic blur fields and blurred images. Specifically, we use the sequences *room1* - *room6* from an existing visual-inertial dataset [19]. These sequences consist of various types of camera motion, which results to a diverse set of blur fields with varying lev-

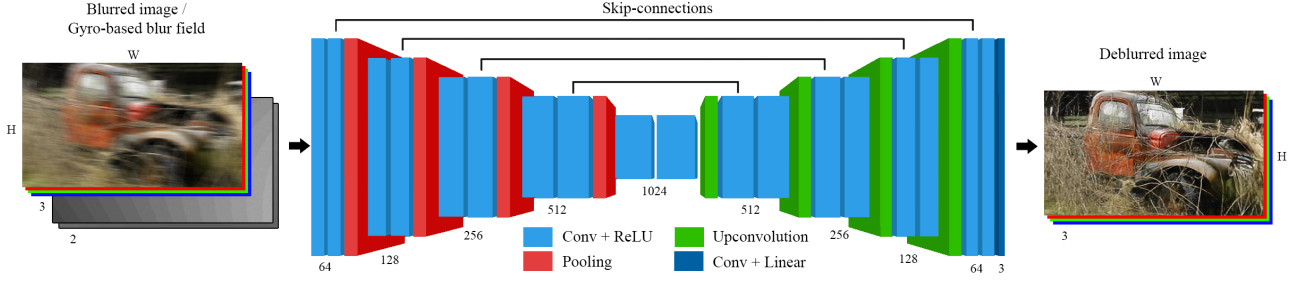


Figure 3. The DeepGyro CNN architecture. All convolutional layers use a 3x3 window (except the last one, which is 1x1). The number of channels is shown below the boxes. Downsampling is 2x2 max pooling with stride 2. Upconvolutional layers consist of upsampling and 2x2 convolution that halves the number of feature channels.

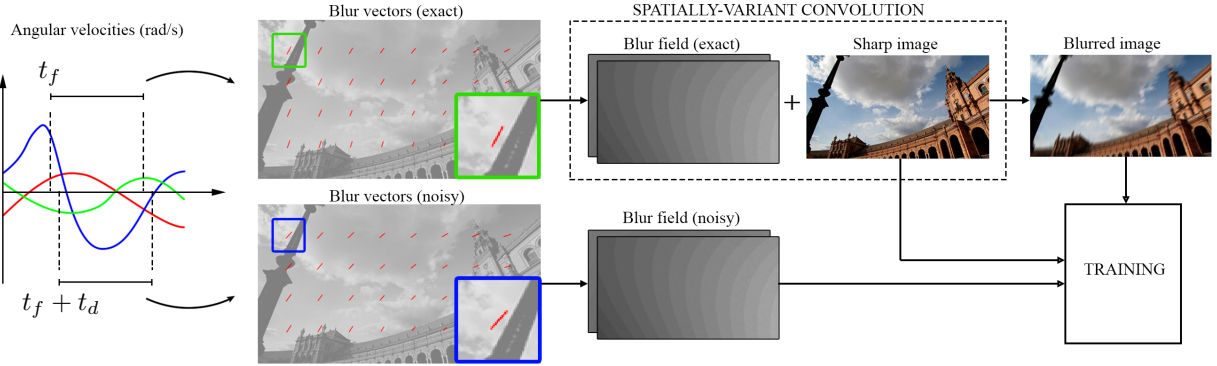


Figure 4. DeepGyro training. For each image, two slightly different blur fields are generated, one that is applied to the sharp image (exact blur, top) and a noisy blur, modelling the IMU derived field (noisy, bottom).

els of spatially-variant motion blur. We also utilize images from the Flickr image collection [8] to cover a wide range of different scene types. With the proposed data generation scheme, it is easy to generate practically unlimited amount of realistic training data. The data generation tool will be made publicly available upon the publication of the paper.

The overview of the data generation scheme is shown in Figure 4. We compute two different blur fields, which we refer as the "exact" and "noisy" blur fields. The exact blur field is used for generating the blurred image. We perform a spatially-variant convolution given a sharp image and blur kernels for every pixel. The noisy blur field, which is slightly different, is provided for the deblurring network as an additional input.

To generate a blur field, we use the approach described in Section 3. The start of the exposure  $t_f$  is selected randomly, which means every blur field is likely to be somewhat different. We set the exposure time  $t_e = 30$  milliseconds. The readout time  $t_r$  is chosen randomly from the range  $[0, 30]$  milliseconds. The zero value corresponds to a global shutter camera. To increase the overall level of motion blur, the angular velocities were first multiplied by 2. However, the maximum blur was limited to 100 pixels.

To simulate temporal misalignment between the camera and gyroscope, we add a small delay  $t_d$  to the start of the

exposure when computing the noisy blur field. The delay is sampled from normal distribution with  $\mu = 0$  and  $\sigma = 0.01$  milliseconds. The translation will also affect the motion blur if the scene is close to the camera. In such case, the blur extents observed by the gyroscope will be somewhat inaccurate. To this end, we multiply the gyroscope readings with a small number  $k \sim N(0, 0.2)$  before computing the noisy blur field. This will mainly affect the blur extents, rather than the direction of the blur.

### 4.3. Training

DeepGyro was trained on 100k images with resolution of  $270 \times 480$  pixels. We used the Adam [11] as the solver. At the beginning, the learning rate was set to 0.00005. After every 10-th epoch, the learning rate was halved. The network was trained for 40 epochs. For comparison, we also trained a blind deblurring network, which we refer to as DeepBlind. In contrast to DeepGyro, it does not take the blur field as input. The network and training details are otherwise identical.

## 5. Experiments

Deblurring performance is evaluated on both synthetically and naturally blurred images. We compare the proposed approaches against DeblurGAN [12] and Mustaniemi

*et al.* [14]. DeblurGAN is a blind deblurring method based on the conditional generative adversarial networks. Similar to our DeepBlind approach, it only takes the blurred image as input. The gyro-based deblurring method [14], referred to as FastGyro, is the closest competitor to our DeepGyro approach. We use a slightly modified version of the original implementation. The blur kernels are estimated for each pixel instead of image patches. This minimizes the artifacts near the edges of the patches.

### 5.1. Synthetic blur

For the quantitative comparison, we add synthetic motion blur and 30 dB Gaussian noise to sharp images [13]. The evaluation metrics include peak-signal-to-noise ratio (PSNR) and structural similarity (SSIM). For fairness, the motion blur is spatially-invariant since DeblurGAN [12] is not designed to handle spatially-variant blur. Note that we also need to generate noisy blur fields for the non-blind methods because the gyroscope readings do not really exist.

Figure 5 shows the deblurring results on a heavily blurred image. DeepBlind and DeepGyro clearly outperform the rest of the methods. Their performance is comparable to each other, although DeepGyro results to a slightly higher PSNR and SSIM values. The average results for all scenes are summarized in Table 1. DeepGyro benefits from the initial blur estimates, especially when there is significant amount of blur.

Figure 6 shows the performance of DeepGyro for increasing levels of motion blur. The method is able to handle extremely strong motion blur. It performs well even when the input blur is not perfect. Figure 7 investigates the effects of blur estimation errors in more detail. Notice that FastGyro [14] is quite sensitive to these errors as there are major ringing artifacts. Another important property of DeepGyro is that it never ruins an already sharp image.

### 5.2. Natural blur

Naturally blurred images were captured with the NVIDIA Shield tablet while simultaneously logging gyroscope at 100 Hz. In this section, we rely on visual assessment since the ground truth sharp images are not available. Figure 8 shows the deblurring results. The resolution of the images is 512 x 512 pixels. DeepGyro performs consistently better than the other methods. In many cases, DeepBlind leaves some parts of the image blurred. FastGyro [14] is able to recover a lot of details but the artifacts reduce the quality of the image. DeblurGAN [12] struggles with strong motion blur. It also seems to produce a grid-like pattern over the image. We also tested our method on a blurred video sequence. Figure 1 (left) shows the result for a single frame with a resolution of 270 x 480 pixels. The deblurring takes around 35 milliseconds on NVIDIA GeForce GTX

1080 GPU. The full video is provided in the supplementary material.

None of the methods is designed for dynamic scene deblurring. Nevertheless, Figure 9 shows a dynamic scene in which a moving car is tracked by the camera. DeepGyro is able to remove most of the blur caused by the camera motion. The car also remains sharp, although a small area around the car is left blurred. This problem is likely due to the fact that the blur does not vary smoothly across the image (as it would in case of camera motion only).

The results are generally quite impressive but there is still room for an improvement. The *entrance* scene in Figure 8 contains bright light sources, which cause some of the pixels to saturate. Consequently, this area is not deblurred. The light streaks also indicate that the blur is somewhat non-linear. This will likely reduce the deblurring performance because such images are not present in the training set. The *flower* scene also shows that a significant translation can cause problems when the scene is close. In this case, it is probable that the gyro-based blur field differs too much from the real blur.

### 5.3. Feature detection and matching

Motion blur degrades the performance of existing feature detectors and descriptors [3]. In this section, we use the proposed methods to improve the robustness against motion blur. Specifically, we use the publicly available implementation of Difference of Gaussian (DoG) detector and SIFT descriptor [27]. The experiment is performed on real-world images with spatially-variant motion blur. The images are shown in Figure 10.

For the evaluation, we need to know the ground truth homography between the images. It defines the mapping of image points in the first and second image given a planar scene. Normally, the homography can be estimated by selecting corresponding points from the images. In this case, the images are blurred, which makes it difficult to select the points accurately. To solve the issue, we adapt the procedure from [14]. The idea is to capture a burst of images while alternating short and long exposure time. The corresponding points are easier to select from the short exposure images, which are sharp but noisy. The blurred images in Figure 10 also suffer from the rolling shutter distortion. Therefore, a homography cannot necessarily perfectly define the mapping of image points. Nevertheless, we concluded that the homographies are sufficiently accurate for this experiment.

To evaluate feature detection, we compute the repeatability, i.e. how well does the detector identify the corresponding image regions. It is well known that the repeatability criteria might favor detectors that return many key-points. To eliminate this issue, we fix the number of detections. The results of the experiment are shown in Figure 10. DeepGyro and DeepBlind clearly outperform the standard



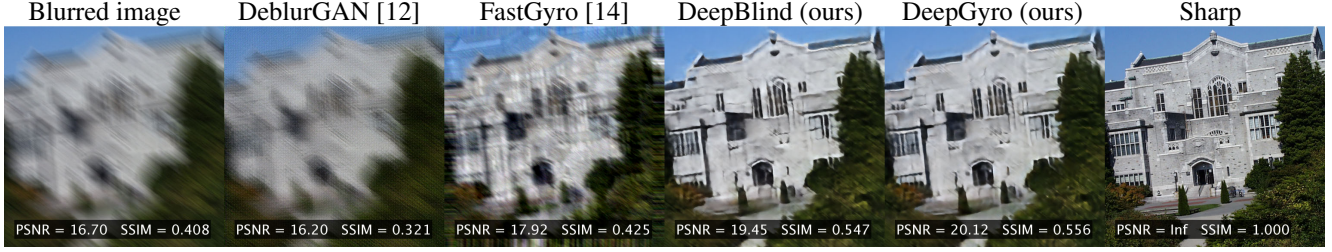


Figure 5. Deblurring results on an image with synthetic linear blur, length 60 pixels. The blur passed to the non-blind methods – FastGyro and DeepGyro – is biased,  $\epsilon = [5, 3]$  pixels is added to the blur vector in the x and y directions, respectively.



Figure 6. DeepGyro performance for increasing levels of blur. Blurred images (green). Results obtained when passing the exact blur  $\epsilon = [0, 0]$  as input (orange), and when the blur vector has an error  $\epsilon = [5, 3]$  pixels (blue). Testing with blur sizes: 10, 40, 60, 80 pixels.



Figure 7. The effects of blur estimation errors on the FastGyro [14] (orange) and DeepGyro (blue). Showing the results when the input is sharp (top row) and when it is blurred (bottom row). The error of the blur is gradually increased  $\epsilon = k * [5, 3]$  pixels, where  $k = 0, 2, 4$ .

detector without deblurring as well as the FastGyro [14].

For the feature matching evaluation, we compute the number of correct matches and the matching score. The nearest neighbour in the descriptors space corresponds to a match. The matching score is the ratio between the number of correct matches and the smaller number of detected features in the pair of images. The results of the experiment are shown in Figure 10. Again, the performance of

DeepGyro and DeepBlind is superior compared to the other approaches.

In this experiment, the performance of DeepGyro and DeepBlind is close to equal. The scene in Figure 10 has a lot of texture, which helps especially the DeepBlind. The information from the gyroscope seems to be redundant when DeepBlind performs well.





Figure 8. Deblurring images blurred by camera motion. From top to bottom: *bridge, tower, church, entrance, flower, posters, office*.

Table 1. Quantitative comparison of deblurring methods on synthetically blurred images: the average PSNR and SSIM metrics for increasing levels of motion blur on the first image from the *graf*, *ubc*, *bikes* and *leuven* image sets [13]. DeepGyro\* - results when the input blur is exact (blur only caused by rotation).

Blur size (pixels)	Blurred image		DeblurGAN [12]		FastGyro [14]		DeepBlind		DeepGyro		DeepGyro*	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
10	23.41	0.792	21.46	0.694	18.55	0.567	29.10	0.920	28.91	0.918	<b>29.34</b>	<b>0.921</b>
20	20.69	0.705	20.23	0.642	19.54	0.611	<b>27.14</b>	0.879	26.93	0.880	27.10	<b>0.880</b>
40	18.64	0.647	18.39	0.593	19.73	0.642	24.32	0.815	24.58	0.821	<b>24.74</b>	<b>0.823</b>
60	17.61	0.617	17.27	0.565	19.18	0.633	21.92	0.746	22.55	0.757	<b>22.66</b>	<b>0.759</b>
80	16.97	0.598	16.70	0.557	18.28	0.605	19.13	0.652	<b>20.20</b>	<b>0.684</b>	19.97	0.681

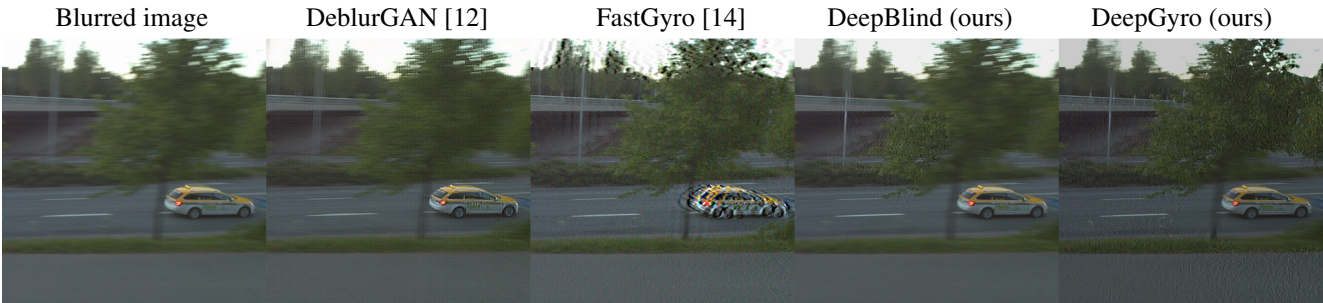


Figure 9. Deblurring a dynamic scene.

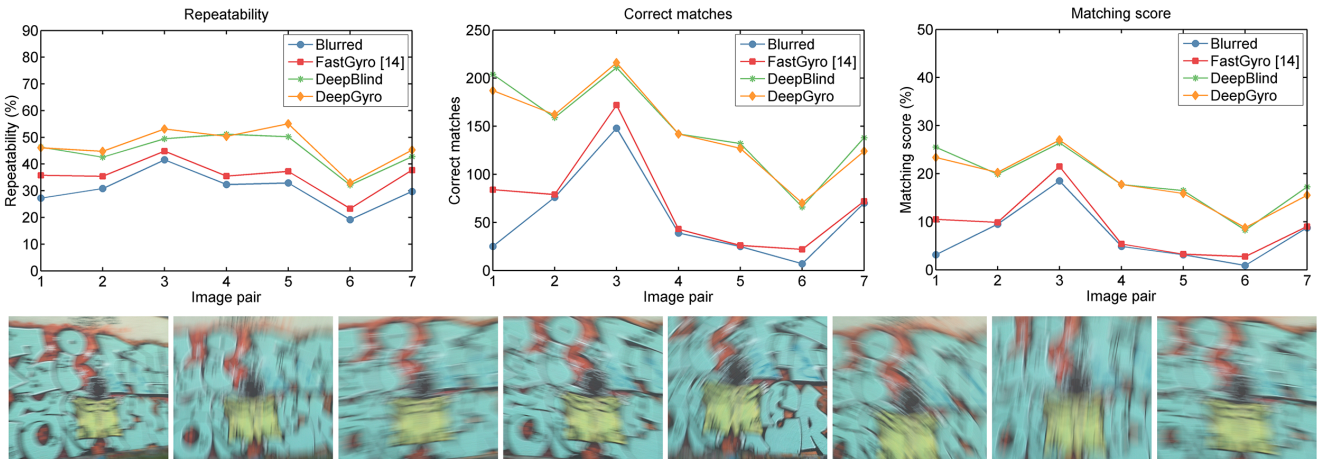


Figure 10. The evaluation of feature detection and matching. Images used in the experiment (bottom). The left-most image is used as the reference. Repeatability scores computed for each image pair (left). The overlap criteria is set to 40 % and the number of detections is fixed to 800. Number of correct nearest neighbour matches (center) and matching scores (right).

## 6. Conclusion

We proposed a deblurring method that is first to pass gyroscope readings to a CNN. The network learns that gyro-based blur estimates are noisy, which allows it to avoid deblurring artifacts common to non-blind deconvolution methods. The evaluation shows that the method handles extreme and spatially-variant motion blur in real-time, unlike existing methods, and that it does not damage images that are sharp. Many of the aforementioned benefits are made possible by the proposed data generation scheme,

which utilizes gyroscope readings to produce realistic training data. Finally, it was demonstrated that the method improves performance of existing feature detectors and descriptors against the motion blur.

## Acknowledgment

The work has been financially supported by the FiDiPro programme of Business Finland and J. Matas was supported by OP VVV MEYS project CZ.02.1.01/0.0/0.0/16\_019/0000765 "Research Center for Informatics".



## References

- [1] M. Aittala and F. Durand. Burst image deblurring using permutation invariant convolutional neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 731–747, 2018.
- [2] H. Chen, J. Gu, O. Gallo, M. Liu, A. Veeraraghavan, and J. Kautz. Reblur2deblur: Deblurring videos via self-supervised learning. In *IEEE International Conference on Computational Photography (ICCP)*, pages 1–9, May 2018.
- [3] S. Gauglitz, T. Höllerer, and M. Turk. Evaluation of interest point detectors and feature descriptors for visual tracking. *International journal of computer vision*, 94(3):335, 2011.
- [4] D. Gong, J. Yang, L. Liu, Y. Zhang, I. D. Reid, C. Shen, A. van den Hengel, and Q. Shi. From motion blur to motion flow: A deep learning solution for removing heterogeneous motion blur. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3806–3815, 2017.
- [5] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.
- [6] S. Hee Park and M. Levoy. Gyro-based multi-image deconvolution for removing handshake blur. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3366–3373, 2014.
- [7] Z. Hu, L. Yuan, S. Lin, and M.-H. Yang. Image deblurring using smartphone inertial sensors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1855–1864, 2016.
- [8] M. J. Huiskes, B. Thomee, and M. S. Lew. New trends and ideas in visual concept detection: the MIR flickr retrieval evaluation initiative. In *Proceedings of the international conference on Multimedia information retrieval*, pages 527–536. ACM, 2010.
- [9] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *CVPR*, 2017.
- [10] N. Joshi, S. B. Kang, C. Lawrence Zitnick, and R. Szeliski. Image deblurring using inertial measurement sensors. *ACM Transactions on Graphics (TOG)*, 29, 07 2010.
- [11] D. P. Kingma and J. Ba. Adam: a method for stochastic optimization. In *International Conference on Learning Representations*, 2015.
- [12] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin, and J. Matas. DeblurGAN: Blind motion deblurring using conditional adversarial networks. *ArXiv e-prints*, 2017.
- [13] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE transactions on pattern analysis and machine intelligence*, 27(10):1615–1630, 2005.
- [14] J. Mustaniemi, J. Kannala, S. Särkkä, J. Matas, and J. Heikkilä. Fast motion deblurring for feature detection and matching using inertial measurements. *arXiv preprint arXiv:1805.08542*, 2018.
- [15] S. Nah, T. H. Kim, and K. M. Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
- [16] T. M. Nimisha, A. K. Singh, and A. N. Rajagopalan. Blur-invariant deep learning for blind-deblurring. In *ICCV*, pages 4762–4770, 2017.
- [17] J. Pan, D. Sun, H. Pfister, and M.-H. Yang. Blind image deblurring using dark channel prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1628–1636, 2016.
- [18] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.
- [19] D. Schubert, T. Goll, N. Demmel, V. Usenko, J. Stueckler, and D. Cremers. The TUM VI benchmark for evaluating visual-inertial odometry. In *International Conference on Intelligent Robots and Systems (IROS)*, October 2018.
- [20] 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.
- [21] A. Solin, S. Cortes, E. Rahtu, and J. Kannala. Inertial odometry on handheld smartphones. In *Proceedings of the International Conference on Information Fusion (FUSION)*, 2018.
- [22] H. Son and S. Lee. Fast non-blind deconvolution via regularized residual networks with long/short skip-connections. In *Computational Photography (ICCP)*, 2017 *IEEE International Conference on*, pages 1–10. IEEE, 2017.
- [23] S. Su, M. Delbracio, J. Wang, G. Sapiro, W. Heidrich, and O. Wang. Deep video deblurring for hand-held cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1279–1288, 2017.
- [24] J. Sun, W. Cao, Z. Xu, and J. Ponce. Learning a convolutional neural network for non-uniform motion blur removal. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 769–777, 2015.
- [25] Y.-W. Tai, H. Du, M. S. 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, 2010.
- [26] D. H. Titterton and J. L. Weston. *Strapdown Inertial Navigation Technology*. The Institution of Electrical Engineers, 2004.
- [27] A. Vedaldi. An open implementation of the SIFT detector and descriptor (2007). *Google Scholar*, 2007.
- [28] R. Wang and D. Tao. Training very deep CNNs for general non-blind deconvolution. *IEEE Transactions on Image Processing*, 27(6):2897–2910, 2018.
- [29] Y. Yan, W. Ren, Y. Guo, R. Wang, and X. Cao. Image deblurring via extreme channels prior. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6978–6986, 2017.
- [30] L. Yuan, J. Sun, L. Quan, and H.-Y. Shum. Image deblurring with blurred/noisy image pairs. In *ACM Transactions on Graphics (TOG)*. ACM, 2007.
- [31] Y. Zhang and K. Hirakawa. Combining inertial measurements with blind image deblurring using distance transform. *IEEE Transactions on Computational Imaging*, 2(3):281–293, 2016.