

# Real-World Blur Dataset for Learning and Benchmarking Deblurring Algorithms

Jaesung Rim<sup>1</sup>, Haeyun Lee<sup>1</sup>, Jucheol Won<sup>2</sup>, and Sunghyun Cho<sup>2</sup>

<sup>1</sup> DGIST, Daegu, Korea

{jsrim,haeyun}@dgist.ac.kr

<sup>2</sup> POSTECH, Pohang, Korea

{jcwon,s.cho}@postech.ac.kr

**Abstract.** Numerous learning-based approaches to single image deblurring for camera and object motion blurs have recently been proposed. To generalize such approaches to real-world blurs, large datasets of real blurred images and their ground truth sharp images are essential. However, there are still no such datasets, thus all the existing approaches resort to synthetic ones, which leads to the failure of deblurring real-world images. In this work, we present a large-scale dataset of real-world blurred images and ground truth sharp images for learning and benchmarking single image deblurring methods. To collect our dataset, we build an image acquisition system to simultaneously capture geometrically aligned pairs of blurred and sharp images, and develop a postprocessing method to produce high-quality ground truth images. We analyze the effect of our postprocessing method and the performance of existing deblurring methods. Our analysis shows that our dataset significantly improves deblurring quality for real-world blurred images.

## 1 Introduction

Images captured in low-light environments such as at night or in a dark room often suffer from motion blur caused by camera shakes or object motions as the camera requires a long exposure time. Such motion blur severely degrades the image quality, and the performance of other computer vision tasks such as object recognition. Thus, image deblurring, a problem to restore a sharp image from a blurred one, has been extensively studied for decades [12, 38, 7, 45, 46, 33, 40, 6].

Recently, several deep learning-based approaches [31, 42, 21, 22] have been proposed and shown a significant improvement. To learn deblurring of real-world blurred images, they require a large-scale dataset of real-world blurred images and their corresponding ground truth sharp images. However, there exist no such datasets so far due to difficulties involved with acquisition of real-world data, which forces the existing approaches to resort to synthetic datasets, e.g., the GoPro dataset [31]. As a result, they do not generalize well to real-world blurred images as will be shown in our experiments.

The main challenge in developing a real-world blur dataset is that the contents of a blurred image and its ground truth sharp image should be geometrically aligned under the presence of blur. This means that the two images should

be taken at the same camera position, which is difficult as the camera must be shaken to take a blurred image. Besides, a real-world blur dataset for image deblurring should satisfy the following requirements. First, the dataset should cover the most common scenarios for camera shakes, i.e., low-light environments where motion blurs most frequently occur. Second, the ground truth sharp images should have as little noise as possible. Lastly, the blurred and ground truth sharp images should be photometrically aligned.

In this paper, we present the first large-scale dataset of real-world blurred images for learning and benchmarking single image deblurring methods, which is dubbed *RealBlur*. Our dataset consists of two subsets sharing the same image contents, one of which is generated from camera raw images, and the other from JPEG images processed by the camera ISP. Each subset provides 4,556 pairs of blurred and ground truth sharp images of 232 low-light static scenes. The blurred images in the dataset are blurred by camera shakes, and captured in low-light environments such as streets at night, and indoor rooms to cover the most common scenarios for motion blurs. To tackle the challenge of geometric alignment, we build an image acquisition system that can simultaneously capture a pair of blurred and sharp images that are geometrically aligned. We also develop a postprocessing method to produce high-quality ground truth images.

With the *RealBlur* dataset, we provide various analyses. We analyze the accuracy of our geometric alignment and its effect on learning of image deblurring. We evaluate existing synthetic datasets as well as ours and seek for the best strategy for training. We also benchmark existing deblurring methods and analyze their performance. Our analysis shows that the *RealBlur* dataset greatly improves the performance of deep learning-based deblurring methods on real-world blurred images. The analysis also shows that networks trained with our dataset can generalize well to dynamic scenes with moving objects.

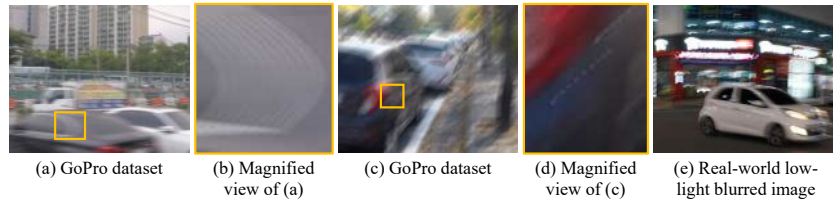
## 2 Related Work

**Single-image deblurring.** Traditional deblurring approaches [12, 38, 7, 45, 46, 33, 40, 6, 24, 25] often model image blur using a convolution operation as:

$$b = k * l + n \quad (1)$$

where  $b$ ,  $l$ , and  $n$  denote a blurry image, a latent image, and additive noise, respectively.  $*$  is a convolution operator, and  $k$  is a blur kernel. Based on this model, previous approaches solve an inverse problem to find  $k$  and  $l$  from  $b$ . Unfortunately, they often fail to handle real-world blurred images because of their restrictive blur model and the ill-posedness of the inverse problem. To deal with more realistic blur, several approaches with extended blur models have been proposed, but their performance is still limited due to the inherent ill-posedness of the inverse problem [18, 44, 8, 14, 17, 19].

Recent deep learning-based approaches [31, 42, 21, 22] overcome such limitations by learning a mapping from a blurry image to its corresponding sharp image from a large collection of data. However, their performance is limited due to the lack of real-world blur datasets. Recently, a few unsupervised learning-based approaches have been proposed, which do not require geometrically-aligned blurred

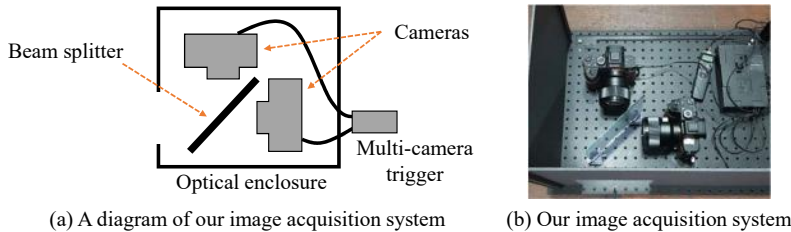


**Fig. 1.** The synthetically blurred images in (a) and (c) in the GoPro dataset [31] are captured in well-lit environments and have unrealistic discontinuous blurs. Blending sharp images cannot mimic saturated light streaks often observed in real-world blurred images like (e) due to the limited dynamic range of sharp images.

and sharp images for learning [28, 27]. However, they are limited to specific domains, e.g., faces and texts, as they rely on generative adversarial networks [13]. **Deblurring datasets.** Several datasets have been proposed along with deblurring methods. However, most of them are designed not for learning but for evaluation of deblurring algorithms. Levin *et al.* [24] proposed a dataset of 32 images blurred by real camera shakes. Sun *et al.* [40] introduced a synthetic dataset generated from 80 natural images and the eight blur kernels. Köhler *et al.* [20] introduced a dataset of 48 blurred images with spatially-varying blur caused by real camera shakes. All these datasets are too small to train neural networks, and unrealistic as they are either synthetically generated or captured in controlled lab environments. Lai *et al.* [23] introduced a dataset of 100 real blurred images for benchmarking deblurring methods. However, their dataset does not provide ground truth sharp images, which are essential for learning image deblurring.

Recently, several synthetic datasets for learning image deblurring have been proposed [31, 30, 32, 39, 51]. To synthetically generate blurred images, they capture sharp video frames using a high-speed camera, and blend them. The resulting images have blurs caused by both spatially-varying camera shakes and object motions. However, due to the extremely short exposure times of the high-speed camera, all the sharp frames were captured in well-lit environments, which are unrealistic for motion blurs to occur. Also, blending sharp frames cannot perfectly mimic the long exposure time of real blurry images because of temporal gaps between adjacent video frames and the limited dynamic range. Thus, networks trained with them do not generalize well to real-world blurry images captured in low-light environments as will be shown in Sec. 5.

**Hybrid imaging.** Our image acquisition system is inspired by previous hybrid imaging approaches. Ben-Ezra and Nayar [2] proposed an hybrid camera system equipped with an additional high-speed low-resolution camera to capture the camera motion. Tai *et al.* [41] extended the approach for spatially-varying blur. Li *et al.* [26] proposed a hybrid camera system for motion deblurring and depth map super-resolution. Yuan *et al.* [47] and Šorel *et al.* [52] capture a pair of noisy and blurred images using exposure bracketing for accurate blur kernel estimation. However, all these approaches are designed for blur kernel estimation, and provide neither high-quality ground-truth images nor sophisticated postprocessing methods like ours.



**Fig. 2.** Our image acquisition system and its diagram.

### 3 Image Acquisition System and Process

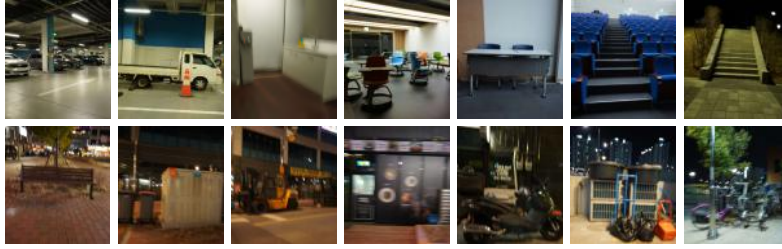
#### 3.1 Image Acquisition System

To capture blurred and sharp images simultaneously, we built a dual camera system (Fig. 2). Our system consists of a beam splitter and two cameras so that the cameras can capture the same scene. The cameras and beam splitter are installed in an optical enclosure to protect them from light coming from outside the viewing direction. One camera captures a blurry image with a low shutter speed, while the other captures a sharp image with a high shutter speed. The two cameras and their lenses are of the same models (Sony A7RM3, Samyang 14mm F2.8 MF). The cameras are synchronized by a multi-camera trigger to capture images simultaneously. Our system is designed to use high-end mirrorless cameras with full-frame sensors and wide-angle lenses based on the following reasons. First, we want to reflect the in-camera processing of conventional cameras into our dataset because blurry JPEG images processed by camera ISPs are more common than raw images. Second, full-frame sensors and wide-angle lenses can gather a larger amount of light than small sensors and narrow-angle lenses so they can more effectively suppress noise. Wide-angle lenses also help avoid defocus blur that may adversely affect learning of motion deblurring.

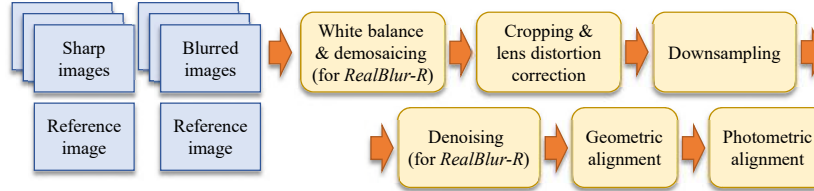
The cameras are physically aligned as much as possible. To evaluate the alignment of the cameras, we conducted stereo calibration [50, 15] and estimated the baseline between the cameras. The estimated baseline is 8.22 mm, which corresponds to disparity of less than four pixels for objects more than 7.8 meters away in the full resolution, and less than one pixel in our final dataset, which contains images downsampled by  $1/4$ .

#### 3.2 Image Acquisition Process

Using our image acquisition system, we captured blurred images of various indoor and outdoor scenes. For each scene, we first captured a pair of two sharp images, referred to as a reference pair, which will be used for geometric and photometric alignment of sharp and blurred images in the postprocessing step. We then captured 20 pairs of blurred and sharp images of the same scene to increase the amount of images and the diversity of camera shakes. For reference pairs, we set the shutter speed to  $1/80$  sec. and adjusted ISO and the aperture size to avoid blur caused by camera shakes. Then, we used the same camera setting for one camera to capture sharp images, while we set the shutter speed of the



**Fig. 3.** Blurred images in the *RealBlur* dataset. Our dataset consists of both dim-lit indoor and outdoor scenes where motion blur commonly occurs.



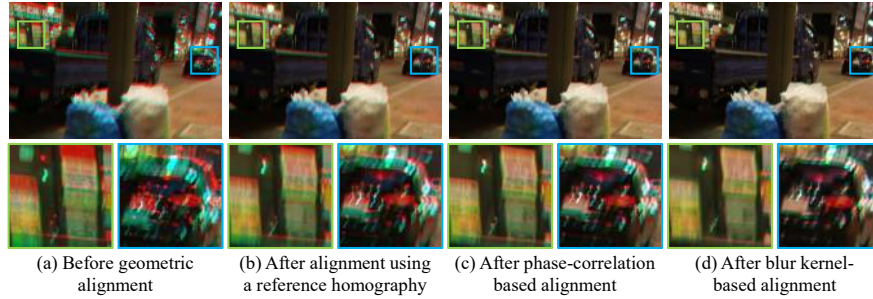
**Fig. 4.** Overall procedure of our postprocessing.

other camera to 1/2 sec. and the ISO value 40 times lower than the reference ISO value to capture blurred images of the same brightness. To capture diverse camera shakes, we simply held our system still for some images, and randomly moved the system for the others. In both cases, blurred images are obtained due to the long exposure time. We captured 4,738 pairs of images of 232 different scenes including reference pairs. We captured all images both in the camera raw and JPEG formats, and generated two datasets: *RealBlur-R* from the raw images, and *RealBlur-J* from the JPEG images. Fig. 3 shows samples of the blurred images in the *RealBlur* dataset.

## 4 Postprocessing

The captured image pairs are postprocessed for noise reduction, and geometric and photometric alignment. Fig. 4 shows an overview of our postprocessing. We first briefly explain the postprocessing procedure for *RealBlur-R*. For each pair of sharp and blurred images, we first apply white balance and demosaicing. For white balance, we use the white balance parameters obtained from the cameras. For demosaicing, we use the adaptive homogeneity-directed demosaicing [16]<sup>3</sup>. As we use a beam splitter and an optical enclosure as well as wide-angle lenses, images have invalid areas along the boundaries that capture outside the beam splitter or inside the optical enclosure. Thus, we crop out such regions. We then correct lens distortions in the cropped images using distortion parameters estimated in a separate calibration step [15]. Then, we downsample the images, and perform denoising to the downsampled sharp image. Finally, we perform geometric and photometric alignment. The sizes of the images from the cameras, after cropping, and after downsampling are  $7952 \times 5304$ ,  $2721 \times 3094$ , and  $680 \times$

<sup>3</sup> We used the *libraw* library for decoding and demosaicing raw images.



**Fig. 5.** Geometric alignment. Each alignment result is shown as a stereo-anaglyph image, where the sharp and blurred images are visualized in red and cyan, respectively, and overlaid to each other. The blurred image has slightly non-uniform blur due to camera shakes so the shapes of cyan light streaks differ across different regions.

773, respectively. For *RealBlur-J*, we follow the same procedure except for white balance, demosaicing, and denoising, as they are performed by camera ISPs. In the following, we explain the downsampling, denoising, and geometric and photometric alignment steps in more detail.

#### 4.1 Downsampling & Denoising

In the downsampling step, we downsample images by  $1/4$  for each axis. The downsampling has three purposes. First, while the image resolutions of recent cameras are very high, even the latest deep learning-based deblurring methods cannot handle such high-resolution images. Second, as we use high ISO values to capture sharp images, they have amplified noise, which can adversely affect training and evaluation of deblurring methods using the sharp images. Downsampling can reduce such noise as it averages nearby pixel intensities. Third, as the alignment of the cameras in our image acquisition system is not perfect, there can exist a small amount of parallax between sharp and blurred images, which can also be effectively reduced by downsampling.

While we reduce noise by downsampling, the downsampled images may still have remaining noise. To further reduce noise, we apply denoising to the sharp images in the denoising step. For each sharp image, we estimate the amount of noise using Chen *et al.*'s method [5]. We then apply the BM3D denoising method [10] setting the noise level parameter to 1.5 times the estimated noise level. We denoise only sharp images. Regarding blurred images, noise is not an issue because it is natural for them to have noise as they are supposed to be captured in low-light conditions, and also because networks trained with noisy blurred images will simply learn both denoising and deblurring.

#### 4.2 Geometric Alignment

Although our image acquisition system has physically well-aligned cameras, there still exists some amount of geometric misalignment (Fig. 5(a)). Furthermore, the positions of the cameras may slightly change over time due to camera shakes. To

address this issue, we conduct a carefully designed geometric alignment process consisting of three steps.

In the first step, we roughly align each blurred and sharp image pair using a homography. As estimating a homography from a blurred and sharp image pair is difficult due to blur, we use a homography estimated from the reference pair corresponding to the target blurred and sharp image pair. For homography estimation, we use the enhanced correlation coefficients method [11] as it is robust to photometric misalignment. Note that geometric alignment using a single homography is possible thanks to the short baseline of our system and the downsampling step, which makes parallax between two images mostly negligible.

Even after alignment using a homography, there can still exist minuscule misalignment between blurred and sharp images due to their different shutter speeds. Specifically, while the multi-camera trigger in our system synchronizes the shutters to open at the same time, they still close at different moments due to their different shutter speeds. Thus, after the shutter of one camera is closed, the other camera still captures incoming lights while moving, causing misalignment between blurred and sharp images. As a result, simply applying the homography of a reference pair results in objects in the sharp image aligned to corners of their corresponding blurry objects in the blurred image, not the centers (Fig. 5(b)).

Thus, in the second step, we estimate the remaining misalignment between each pair of blurred and sharp images, and align them. To this end, we use a phase correlation-based approach [35] that can robustly estimate a similarity transform under the presence of blur (see our supplementary material for the analysis about its robustness to camera shakes). Fig. 5(c) shows an example of the phase correlation-based alignment, where the red and cyan light reflections are better aligned so that they appear brighter.

The phase correlation-based alignment, however, cannot align the contents in the blurred and sharp images with respect to their centers. Thus, in the third step, we align the blurred and sharp images to match their centers. Our third step is inspired by traditional blur model-based deblurring approaches. Traditional blur model-based approaches such as [37, 4, 9] often align images or blur kernels with respect to the centers of mass, or centroids, of blur kernels to align their deblurring results with blurry input images. Following such approaches, we align images to match the center of an object in a sharp image with the center of its corresponding object in a blurred image in an additional alignment step for each pair of blurry and sharp images. To this end, we estimate a blur kernel of the blurred image using its corresponding sharp image assuming that the scene is static and camera shake is nearly spatially-invariant. Following conventional blur kernel estimation methods [7, 46, 6], we estimate a blur kernel  $k$  by minimizing the following energy function:

$$E(k) = \|k * \nabla s - \nabla b\|^2 + \lambda \|\nabla k\|^2 \quad (2)$$

where  $s$  and  $b$  are sharp and blurred images, respectively, and  $\nabla$  is a gradient operator.  $\lambda$  is the regularization weight, which we set  $\lambda = 10^3$  in our experiment. Then, we compute the centroid of the estimated blur kernel  $k$ , and align the sharp

image by shifting it according to the centroid (Fig. 5(d)). As will be shown in Sec. 5.1, the centroid-based alignment effectively reduces the receptive field size required for deblurring, and enables effective learning of deblurring.

Note that each step in our geometric alignment is essential for accurate alignment as they provide different characteristics. The first step based on the reference pair can align images using a homography but is less accurate. The second step improves the accuracy of alignment in the presence of blur, but is restricted to a similarity transform. The third step can accurately align images based on the centroid while being restricted to translation. The effect of each step will be analyzed in Sec. 5.1.

### 4.3 Photometric Alignment

Although we use cameras and lenses of the same models, their images may have slight intensity difference. To resolve this, we perform photometric alignment based on a linear model following [3, 34, 1]. Specifically, for geometrically aligned sharp and blurred images  $s$  and  $b$ , we photometrically align  $s$  to  $b$  by applying a linear transform  $\alpha s + \beta$  so that  $\alpha s + \beta \approx b$ . The coefficients  $\alpha$  and  $\beta$  are difficult to estimate from  $s$  and  $b$  due to the blur in  $b$ . Thus, we estimate them from the reference pair corresponding to  $s$  and  $b$ . Specifically,  $\alpha$  and  $\beta$  are estimated as  $\alpha = \sigma_1/\sigma_2$  and  $\beta = \mu_1 - \alpha\mu_2$  where  $\sigma_1$  and  $\sigma_2$  are the standard deviations of the reference images, and  $\mu_1$  and  $\mu_2$  are their means. To process color images, we apply the photometric alignment to each color channel independently.

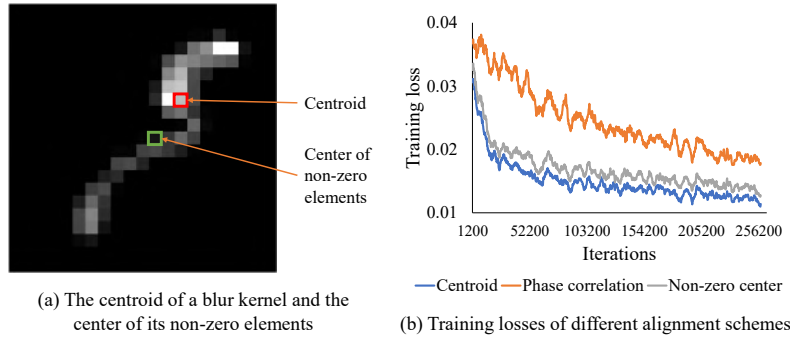
## 5 Experiments

In this section, we analyze our dataset and verify its effect on image deblurring. We also benchmark existing deblurring algorithms on real-world blurry images, and study the effect of our dataset. Additional examples and analyses, e.g., the distribution of blur sizes in our dataset, and the generalization ability to images captured by other cameras, can be found in the supplementary material. All data including the dataset is available on our project webpage<sup>4</sup>.

**Datasets and evaluation measure.** For the benchmark, we randomly select 182 scenes from *RealBlur-R* and *RealBlur-J* as our training sets and the remaining 50 scenes as our test sets. Each training set consists of 3,758 image pairs including 182 reference pairs, while each test set consists of 980 image pairs without reference pairs. We include reference pairs in our training sets so that networks can learn the identity mapping for sharp images. Besides *RealBlur*, we also consider two existing deblurring datasets: GoPro [31], and Köhler *et al.*'s [20]. The GoPro dataset, which is the most widely used by recent deep learning-based approaches, is a synthetic dataset generated by blending sharp video frames captured by a high-speed camera. The GoPro dataset provides 2,103 and 1,111 pairs of blurred and sharp images for its training and test sets. Köhler *et al.*'s dataset is a small-scale set of images with real camera shakes, which are captured in a controlled lab environment. We also consider another purely synthetic dataset, which is generated from the BSD500 segmentation dataset [29] as

<sup>4</sup> <http://cg.postech.ac.kr/research/realblur/>





**Fig. 6.** Training losses of different alignment methods.

follows. For each image in BSD500, we randomly generate 40 synthetic uniform motion blur kernels using Schmidt *et al.*'s method [36] and convolve the image with them to obtain 20,000 blurred images. The sharp images and the blurred images are then aligned with respect to the centroid of the blur kernels. We refer to this dataset as BSD-B in the remainder of this section.

Deblurring may produce misaligned results even when blur is successfully removed. To deal with such misalignment when measuring the quality of deblurring results against ground truth images, we adopt a similar approach to Köhler *et al.* [20] in all the experiments. We first aligns a deblurring result to its ground truth sharp image using a homography estimated by the enhanced correlation coefficients method [11], and computes PSNRs or SSIMs [43].

### 5.1 Analysis on Geometric Alignment

**Effect of geometric alignment.** We analyze the effect of geometric alignment with respect to the centroids of blur kernels on the learning of deblurring. Restoring a sharp pixel requires information from nearby pixels in a blurred image, which sets a lower bound for the receptive field size required for deblurring. Alignment using the centroid of a blur kernel can effectively reduce the required receptive field size and ease the training of networks while visually matching the centers of blurry image contents and their corresponding sharp contents. Another possibly more optimal approach to reducing the required receptive field size is to align an image to the center of the non-zero elements of its blur kernel as it is the closest point to all non-zero kernel elements (Fig. 6(a)). We refer to the center of non-zero elements as the non-zero center for brevity. However, we found that this approach is less effective than using the centroids as discussed below. While it is unclear why, we conjecture that it is because the centroid is the most central position in terms of information amount where we can utilize information of nearby pixels most effectively.

To verify the effectiveness of the centroid-based alignment, we conduct a simple experiment. We generate three differently aligned sets from BSD-B: aligned using translation estimated by the phase correlation [35], aligned to the non-zero centers, and aligned to the centroids. We train SRN-DeblurNet [42], which is a state-of-the-art deep learning-based approach, with the three sets separately, and

**Table 1.** Displacement error of variants of our geometric alignment process.

Geometric alignment methods	Error (pixels)
Reference homography	4.9454
Reference homography+blur kernel	0.8701
Reference homography+phase correlation+blur kernel (ours)	0.8058

compare their performance on Köhler *et al.*'s dataset [20]. The average PSNR values of the phase correlation-, non-zero center-, and centroid-based datasets are 26.48, 27.80, and 28.07 dBs, respectively. Moreover, the centroid-based alignment also results in the most efficient training as shown in Fig. 6(b). This result proves that the alignment based on the centroids of blur kernels in our postprocessing is essential for effective learning of image deblurring.

**Geometric alignment accuracy.** Our geometric alignment assumes a couple of assumptions. First, a pair of blurred and sharp images can be aligned with a single homography. Second, blurred images have nearly uniform blur so that images can be aligned using a single blur kernel. As violation of either of them can degrade the accuracy of our geometric alignment, we verify whether the resulting dataset is accurately aligned. As there are no ground truth alignment, we indirectly compute the average displacement error for each image pair as follows. For each pair of blurred and sharp images that are aligned, we first divide them into a  $2 \times 2$  regular grid. For each grid cell, we estimate a local blur kernel solving Eq. (2). Then, we compute the centroid of the estimated blur kernel. If the blurred and sharp images cannot be aligned using a single homography, or the blurred image has spatially-varying blur, the centroid will be off center of the blur kernel. Thus, we compute the displacement between the centroid and the image center of the local blur kernel as displacement error.

Table 1 shows that the average displacement error of our dataset after our geometric alignment is only less than 1 pixel. The table also shows the average displacement error of other variants of our geometric alignment process to verify the effect of each component. As shown in the table, image pairs aligned using homographies from reference pairs have large displacement error due to the different shutter speeds (1st row). It also shows that the blur kernel-based alignment significantly reduces error (2nd row), and that phase correlation-based alignment further improves the accuracy (3rd row).

As we can estimate local blur kernels, we may directly use them for geometric alignment. For example, we can compute the centroids of local blur kernels estimated from a  $2 \times 2$  grid, then derive a homography from them instead of global translation to align images. However, we found that this approach is less reliable for our dataset and causes larger error because blur kernel estimation is unreliable on textureless image regions or image regions with saturated pixels. Thus, we designed our geometric alignment process to estimate a single global blur kernel that can be more reliably estimated.

## 5.2 Benchmark

**Dataset comparison and training strategy.** Before benchmarking existing deblurring approaches, we first compare the performance of our datasets with

**Table 2.** Performance comparison of different datasets. We trained a deblurring network of Tao *et al.* [42] using different training sets and measured its performance.

Training sets			Pre-trained	Test sets (PSNR/SSIM)		
<i>RealBlur-R</i>	GoPro	BSD-B		<i>RealBlur-R</i>	Köhler	GoPro
	✓			35.66/0.9472	26.79/0.7963	30.72/0.9074
		✓		34.96/0.9132	28.07/0.8259	29.01/0.8768
✓				36.47/0.9515	24.72/0.7422	23.99/0.7675
✓	✓			38.47/0.9632	26.96/0.7991	30.02/0.8946
✓		✓		38.62/0.9649	27.99/0.8249	29.02/0.8774
✓	✓	✓		38.58/0.9646	28.00/0.8241	29.93/0.8931
✓			✓	38.73/0.9646	26.38/0.7942	26.56/0.8422
✓	✓		✓	38.65/0.9646	27.04/0.8017	30.53/0.9045
✓		✓	✓	38.71/0.9657	28.18/0.8294	29.22/0.8824
✓	✓	✓	✓	38.65/0.9652	28.14/0.8311	30.30/0.9006

**Table 3.** Performance comparison of different datasets. We trained a deblurring network of Tao *et al.* [42] using different training sets and measured its performance.

Training sets			Pre-trained	Test sets (PSNR/SSIM)		
<i>RealBlur-J</i>	GoPro	BSD-B		<i>RealBlur-J</i>	Köhler	GoPro
	✓			28.56/0.8674	26.79/0.7963	30.72/0.9074
		✓		28.68/0.8675	28.07/0.8259	29.01/0.8768
✓				31.02/0.8987	26.57/0.7986	26.68/0.8403
✓	✓			31.21/0.9018	26.94/0.8044	29.91/0.8923
✓		✓		31.30/0.9058	27.88/0.8249	28.97/0.8785
✓	✓	✓		31.37/0.9063	27.74/0.8229	29.90/0.8926
✓			✓	31.32/0.9070	26.77/0.8044	27.18/0.8603
✓	✓		✓	31.40/0.9078	27.13/0.8113	30.46/0.9034
✓		✓	✓	31.44/0.9105	28.06/0.8319	29.21/0.8842
✓	✓	✓	✓	31.38/0.9091	27.82/0.8260	30.30/0.9004

other datasets, and seek for the best strategy for training deblurring networks. Specifically, we prepare differently trained models of a deblurring network using several possible combinations of different training sets including ours. Then, we investigate their performance on different test sets. For evaluation, we use SRN-DeblurNet [42]. As a pre-trained version of SRN-DeblurNet trained with the GoPro dataset is already available, we also include it in our experiment. We refer the readers to the supplementary material for details about training.

Tables 2 and 3 show the performance of different combinations of the training sets on different test sets. The column ‘Pre-trained’ indicates whether the network is trained from the pre-trained weights using the GoPro dataset or from scratch. The tables show that the GoPro dataset (1st rows in Tables 2 and 3) achieves lower performance on the *RealBlur* test sets compared to the other combinations in general, which proves that the GoPro dataset is not realistic enough to cover real-world blurred images. The BSD-B dataset (2nd rows in Tables 2 and 3) also achieves low performance on the *RealBlur* test sets, but high performance on Köhler *et al.*’s test set, which is possibly because Köhler *et al.*’s dataset is close to synthetic as its images are captured in a controlled lab environment. On the other hand, our training sets (3rd rows in Tables 2 and 3) achieve higher performance on the *RealBlur* test sets, which validates the necessity of real-world blur training data.

**Table 4.** Benchmark of state-of-the-art deblurring methods on real-world blurred images. **Purple**: traditional optimization-based methods. **Black**: deep learning-based methods. **Blue\***: models trained with our dataset. Methods are sorted in the descending order with respect to PSNR.

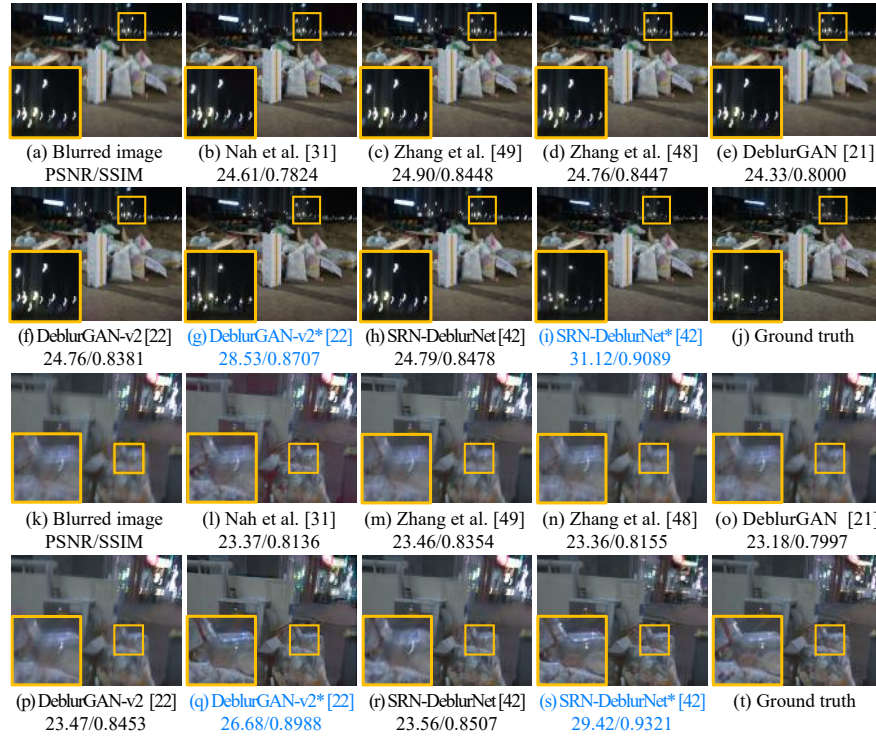
<i>RealBlur-J</i>		<i>RealBlur-R</i>	
Methods	PSNR/SSIM	Methods	PSNR/SSIM
SRN-DeblurNet* [42]	31.38/0.9091	SRN-DeblurNet* [42]	38.65/0.9652
DeblurGAN-v2* [22]	29.69/0.8703	DeblurGAN-v2* [22]	36.44/0.9347
DeblurGAN-v2 [22]	28.70/0.8662	Zhang <i>et al.</i> [49]	35.70/0.9481
SRN-DeblurNet [42]	28.56/0.8674	SRN-DeblurNet [42]	35.66/0.9472
Zhang <i>et al.</i> [49]	28.42/0.8596	Zhang <i>et al.</i> [48]	35.48/0.9466
DeblurGAN [21]	27.97/0.8343	DeblurGAN-v2 [22]	35.26/0.9440
Nah <i>et al.</i> [31]	27.87/0.8274	Xu <i>et al.</i> [46]	34.46/0.9368
Zhang <i>et al.</i> [48]	27.80/0.8472	Pan <i>et al.</i> [33]	34.01/0.9162
Pan <i>et al.</i> [33]	27.22/0.7901	DeblurGAN [21]	33.79/0.9034
Xu <i>et al.</i> [46]	27.14/0.8303	Hu <i>et al.</i> [18]	33.67/0.9158
Hu <i>et al.</i> [18]	26.41/0.8028	Nah <i>et al.</i> [31]	32.51/0.8406

The tables also show that using multiple training sets together tends to achieve higher performance on multiple test sets, as it increases the amount of training data and the range of image contents. Among different combinations, we found that training with all datasets and pre-trained weights achieves relatively good performance on all test sets. Thus, we use it as our default training strategy in the rest of this section.

**Benchmarking deblurring methods.** We then benchmark state-of-the-art deblurring methods including both traditional optimization-based [46, 33, 18] and recent deep learning-based approaches [31, 49, 48, 21, 22, 42] using our test sets. For all the deep learning-based approaches, we use pre-trained models provided by the authors. For DeblurGAN-v2 [22] and SRN-DeblurNet [42], we also include models trained with our training strategy.

Table 4 shows a summary of the benchmark. In the benchmark, the traditional methods achieve relatively low PSNR and SSIM values both for the *RealBlur-R* and *RealBlur-J* test sets, often failing to estimate correct blur kernels. Such traditional approaches are known to be vulnerable to noise, saturated pixels, and non-uniform blur, which are common in real low-light images. On the other hand, the deep learning-based methods are more successful in terms of both PSNR and SSIM, as they are more robust to noise and non-uniform blur. The deep learning-based approaches trained with our training sets show the best performance proving the benefits of training with real low-light blurred images.

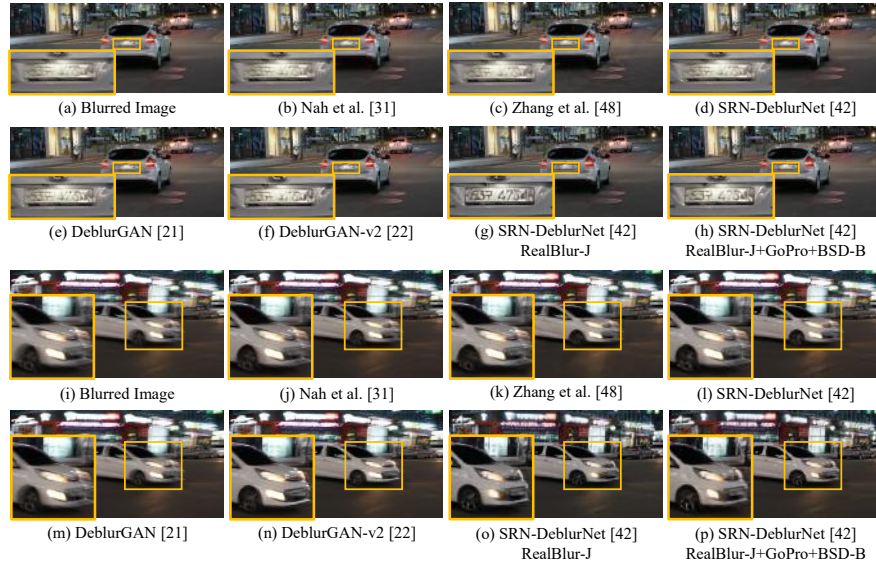
Fig. 7 shows a qualitative comparison of the deep learning-based methods in Table 4. All the models trained without real-world blurred images fail to restore light streaks as well as other image details. On the other hand, the models trained with our datasets show better restored results. The results of SRN-DeblurNet [42] trained with our dataset in Fig. 7(i) and (s) show accurately restored image details. The results of DeblurGAN-v2 [22] trained with our dataset in Fig. 7(g) and (q) also show better restored details than the others, while slightly worse than those of SRN-DeblurNet trained with our datasets.



**Fig. 7.** Qualitative comparison of different deblurring methods on the *RealBlur-J* and *RealBlur-R* test sets. (a)-(j) Deblurring results using *RealBlur-J*. (k)-(t) Deblurring results using *RealBlur-R*. For visualization, the examples of *RealBlur-R* are gamma-corrected. Methods marked with ‘\*’ in blue are trained with our datasets.

**Dynamic scenes.** Our dataset consists of static scenes without moving objects. Thus, one natural question that follows is whether networks trained with our dataset can handle dynamic scenes with moving objects. To answer the question, we investigate the performance of our dataset on dynamic scenes. To this end, we collected a set of real blurred images with moving objects without ground truth sharp images. We used a camera of a different model (Sony A7M2) and different lenses (SEL85F18, SEL1635Z) instead of our image acquisition system to collect images. Then, we perform qualitative evaluation of the performance of deep learning-based methods trained with different training sets.

Fig. 8 shows results of deep learning-based methods with different training sets. The blurred images in Fig. 8 have spatially-varying blurs caused by object motions. For all the methods in this experiment, we use pre-trained models provided by the authors unless specified. While *RealBlur-J* does not have any moving objects, the results in Fig. 8(g) and (o) show that the networks trained only with *RealBlur-J* can successfully restore sharp images. Moreover, the networks trained with *RealBlur-J* can produce better results than the networks trained only with the GoPro dataset, even though the GoPro dataset includes



**Fig. 8.** Qualitative comparison of different methods on images of dynamic scenes.

a large number of dynamic scenes. We refer the readers to the supplementary material for more examples.

## 6 Conclusion

In this paper, we presented the *RealBlur* dataset, which is the first large-scale real-world blur dataset for learning image deblurring. To collect dataset, we built an image acquisition system that can simultaneously capture a pair of blurred and sharp images. We developed a postprocessing method to produce high-quality ground truth images, and analyzed the effect and accuracy of its geometric alignment. Our experiments showed that the *RealBlur* dataset can greatly improve the performance of deep learning-based deblurring approaches on real-world blurred images by camera shakes and moving objects.

**Limitations and future work.** Our *RealBlur* dataset consists of static scenes without moving objects. While we demonstrated that neural networks trained with *RealBlur* can deal with dynamic scenes qualitatively, a dataset of dynamic scenes is essential for quantitative evaluation of dynamic scene deblurring. Although we used high-end mirrorless cameras to collect real-world blurred images, a much larger number of users use smartphone cameras. Thus, collecting a dataset for such low-end cameras would be an interesting future work. Our work can provide a basis for developing deblurring methods for real-world blurred images. It would also be interesting future work to develop a more realistic generative model for synthesizing blurry images, which can be used for learning image deblurring, and the *RealBlur* dataset can be used as a basis for it.

**Acknowledgement:** This work was supported by Samsung Research Funding & Incubation Center of Samsung Electronics under Project Number SRFC-IT1801-05.

## References

1. Abdelhamed, A., Lin, S., Brown, M.S.: A high-quality denoising dataset for smart-phone cameras. In: CVPR (June 2018)
2. Ben-Ezra, M., Nayar, S.: Motion deblurring using hybrid imaging. In: CVPR. pp. 657–664 (2003)
3. Cai, J., Zeng, H., Yong, H., Cao, Z., Zhang, L.: Toward real-world single image super-resolution: A new benchmark and a new model. In: ICCV (October 2019)
4. Chakrabarti, A.: A neural approach to blind motion deblurring. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) Computer Vision – ECCV 2016. pp. 221–235. ECCV, Cham (2016)
5. Chen, G., Zhu, F., Ann Heng, P.: An efficient statistical method for image noise level estimation. In: ICCV (December 2015)
6. Cho, S., Lee, S.: Convergence analysis of map based blur kernel estimation. In: ICCV. pp. 4818–4826 (Oct 2017)
7. Cho, S., Lee, S.: Fast motion deblurring. *ACM Trans. Graph.* **28**(5), 145:1–145:8 (Dec 2009)
8. Cho, S., Wang, J., Lee, S.: Handling outliers in non-blind image deconvolution. In: ICCV (2011)
9. Cho, T.S., Paris, S., Horn, B.K.P., Freeman, W.T.: Blur kernel estimation using the radon transform. In: CVPR (2011)
10. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising by sparse 3-d transform-domain collaborative filtering. *TIP* **16**(8), 2080–2095 (Aug 2007)
11. Evangelidis, G.D., Psarakis, E.Z.: Parametric image alignment using enhanced correlation coefficient maximization. *TPAMI* **30**(10), 1858–1865 (Oct 2008)
12. Fergus, R., Singh, B., Hertzmann, A., Roweis, S.T., Freeman, W.T.: Removing camera shake from a single photograph. *ACM Trans. Graph.* **25**(3), 787–794 (Jul 2006)
13. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (eds.) *Advances in Neural Information Processing Systems 27*, pp. 2672–2680. Curran Associates, Inc. (2014), <http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf>
14. Gupta, A., Joshi, N., Zitnick, L., Cohen, M., Curless, B.: Single image deblurring using motion density functions. In: ECCV. pp. 171–184 (2010)
15. Heikkila, J., Silven, O.: A four-step camera calibration procedure with implicit image correction. In: CVPR. p. 1106. CVPR '97, IEEE Computer Society, USA (1997)
16. Hirakawa, K., Parks, T.W.: Adaptive homogeneity-directed demosaicing algorithm. *TIP* **14**(3), 360–369 (March 2005)
17. Hirsch, M., Schuler, C.J., Harmeling, S., Schölkopf, B.: Fast removal of non-uniform camera shake. In: ICCV. pp. 463–470 (2011)
18. Hu, Z., Cho, S., Wang, J., Yang, M.: Deblurring low-light images with light streaks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **40**(10), 2329–2341 (2018)
19. Kim, T.H., Lee, K.M.: Segmentation-free dynamic scene deblurring. In: CVPR. pp. 2766–2773 (June 2014)
20. Köhler, R., Hirsch, M., Mohler, B., Schölkopf, B., Harmeling, S.: Recording and playback of camera shake: benchmarking blind deconvolution with a real-world database. In: ECCV. pp. 27–40 (2012)

21. Kupyn, O., Budzan, V., Mykhailych, M., Mishkin, D., Matas, J.: DeblurGAN: Blind motion deblurring using conditional adversarial networks. In: CVPR (June 2018)
22. Kupyn, O., Martyniuk, T., Wu, J., Wang, Z.: DeblurGAN-v2: Deblurring (orders-of-magnitude) faster and better. In: ICCV (October 2019)
23. Lai, W.S., Huang, J.B., Hu, Z., Ahuja, N., Yang, M.H.: A comparative study for single image blind deblurring. In: CVPR (June 2016)
24. Levin, A., Weiss, Y., Durand, F., Freeman, W.T.: Understanding and evaluating blind deconvolution algorithms. In: CVPR. pp. 1964–1971 (2009)
25. Levin, A., Weiss, Y., Durand, F., Freeman, W.T.: Efficient marginal likelihood optimization in blind deconvolution. In: CVPR. pp. 2657–2664 (2011)
26. Li, F., Yu, J., Chai, J.: A hybrid camera for motion deblurring and depth map super-resolution. In: CVPR (2008)
27. Lu, B., Chen, J.C., Chellappa, R.: Unsupervised domain-specific deblurring via disentangled representations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
28. Madam Nimisha, T., Sunil, K., Rajagopalan, A.N.: Unsupervised class-specific deblurring. In: Proceedings of the European Conference on Computer Vision (ECCV) (September 2018)
29. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: ICCV. vol. 2, pp. 416–423 (July 2001)
30. Nah, S., Baik, S., Hong, S., Moon, G., Son, S., Timofte, R., Mu Lee, K.: Ntire 2019 challenge on video deblurring and super-resolution: Dataset and study. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops (June 2019)
31. Nah, S., Hyun Kim, T., Mu Lee, K.: Deep multi-scale convolutional neural network for dynamic scene deblurring. In: CVPR (July 2017)
32. Noroozi, M., Chandramouli, P., Favaro, P.: Motion deblurring in the wild. In: Roth, V., Vetter, T. (eds.) Pattern Recognition. pp. 65–77. Springer International Publishing, Cham (2017)
33. Pan, J., Sun, D., Pfister, H., Yang, M.H.: Blind image deblurring using dark channel prior. In: CVPR. pp. 1628–1636 (2016)
34. Plotz, T., Roth, S.: Benchmarking denoising algorithms with real photographs. In: CVPR (July 2017)
35. Reddy, B.S., Chatterji, B.N.: An fft-based technique for translation, rotation, and scale-invariant image registration. TIP **5**(8), 1266–1271 (Aug 1996)
36. Schmidt, U., Jancsary, J., Nowozin, S., Roth, S., Rother, C.: Cascades of regression tree fields for image restoration. TPAMI **38**(4), 677–689 (April 2016). <https://doi.org/10.1109/TPAMI.2015.2441053>
37. Schuler, C.J., Hirsch, M., Harmeling, S., Schölkopf, B.: Learning to deblur. TPAMI **38**(7), 1439–1451 (July 2016)
38. Shan, Q., Jia, J., Agarwala, A.: High-quality motion deblurring from a single image. ACM Trans. Graph. **27**(3), 73:1–73:10 (Aug 2008)
39. Su, S., Delbracio, M., Wang, J., Sapiro, G., Heidrich, W., Wang, O.: Deep video deblurring for hand-held cameras. In: CVPR. pp. 237–246 (July 2017)
40. Sun, L., Cho, S., Wang, J., Hays, J.: Edge-based blur kernel estimation using patch priors. In: ICCP (2013)
41. Tai, Y.W., Du, H., Brown, M.S., Lin, S.: Image/video deblurring using a hybrid camera. In: CVPR (2008)



42. Tao, X., Gao, H., Shen, X., Wang, J., Jia, J.: Scale-recurrent network for deep image deblurring. In: CVPR (June 2018)
43. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. *TIP* **13**(4), 600–612 (April 2004). <https://doi.org/10.1109/TIP.2003.819861>
44. Whyte, O., Sivic, J., Zisserman, A., Ponce, J.: Non-uniform deblurring for shaken images. In: CVPR. pp. 491–498 (2010)
45. Xu, L., Jia, J.: Two-phase kernel estimation for robust motion deblurring. In: ECCV (2010)
46. Xu, L., Zheng, S., Jia, J.: Unnatural L0 sparse representation for natural image deblurring. In: CVPR (2013)
47. Yuan, L., Sun, J., Quan, L., Shum, H.: Image deblurring with blurred/noisy image pairs. In: SIGGRAPH (2007)
48. Zhang, H., Dai, Y., Li, H., Koniusz, P.: Deep stacked hierarchical multi-patch network for image deblurring. In: CVPR (June 2019)
49. Zhang, J., Pan, J., Ren, J., Song, Y., Bao, L., Lau, R.W., Yang, M.H.: Dynamic scene deblurring using spatially variant recurrent neural networks. In: CVPR (June 2018)
50. Zhang, Z.: A flexible new technique for camera calibration. *TPAMI* **22**(11), 1330–1334 (Nov 2000)
51. Zhou, S., Zhang, J., Zuo, W., Xie, H., Pan, J., Ren, J.S.: Davanet: Stereo deblurring with view aggregation. In: CVPR. pp. 10988–10997 (June 2019)
52. Šorel, M., Šroubek, F.: Space-variant deblurring using one blurred and one under-exposed image. In: ICIP. pp. 157–160 (2009)