Master's Thesis
석사 학위논문

Real-World Blur Dataset for Learning and Benchmarking Deblurring Algorithms

Jaesung Rim(임 재 성 任 宰 成)

Department of
Information & Communication Engineering

DGIST

2020

Real-World Blur Dataset for Learning and Benchmarking Deblurring Algorithms

Advisor: Professor Hoon Sung Chwa
Co-advisor: Professor Sunghyun Cho

by

Jaesung Rim
Department of Information & Communication Engineering
DGIST

A thesis submitted to the faculty of DGIST in partial fulfillment of the requirements for the degree of Master of Science in the Department of Information & Communication Engineering. The study was conducted in accordance with Code of Research Ethics[1]

05. 22. 2020

Approved by

Professor Hoon Sung Chwa (signature)
(Advisor)

Professor Sunghyun Cho (signature)
(Co-Advisor)

[1] Declaration of Ethical Conduct in Research: I, as a graduate student of DGIST, hereby declare that I have not committed any acts that may damage the credibility of my research. These include, but are not limited to: falsification, thesis written by someone else, distortion of research findings or plagiarism. I affirm that my thesis contains honest conclusions based on my own careful research under the guidance of my thesis advisor.

Real-World Blur Dataset for Learning and Benchmarking Deblurring Algorithms

Jaesung Rim

Accepted in partial fulfillment of the requirements for the degree of Master of Science.

05. 22. 2020

Head of Committee	Prof. Hoon Sung Chwa	(signature)
Committee Member	Prof. Sunghyun Cho	(signature)
Committee Member	Prof. Sunghoon Im	(signature)

MS/ES
201822022

임 재 성. Jaesung Rim. Real-World Blur Dataset for Learning and Benchmarking Deblurring Algorithms. Department of Information & Communication Engineering. 2020. 29p. Advisors Prof. Hoon Sung Chwa, Co-Advisors Prof. Sunghyun Cho

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 their corresponding sharp images captured in low-light environments for learning and benchmarking single image deblurring methods. To collect our dataset, we build an image acquisition system to simultaneously capture a geometrically aligned pair of blurred and sharp images, and develop a post-processing method to further align images geometrically and photometrically. We analyze the effect of our post-processing step, and the performance of existing learning-based deblurring methods. Our analysis shows that our dataset significantly improves deblurring quality for real-world low-light images.

Keywords: Computational Photography, Deblurring, Low-level Vision, Datasets and Evaluation

List of Contents

List of Tables

List of Figures

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 longer exposure time to collect an enough amount of light. Such motion blur severely degrades the image quality, and also the performance of other computer vision tasks such as object recognition. Thus, image deblurring, which is a problem to restore a sharp image from a blurred one, has been extensively studied for decades in the computer vision and image processing fields [11, 33, 6, 40, 41, 28, 35, 5, 26, 37].

With the advent of deep learning, several deep learning-based deblurring approaches [26, 37, 19, 20] have been proposed and shown a significant improvement. To learn image deblurring of real-world low-light photographs, they require a large-scale dataset that consists 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 blurred images and their corresponding sharp images, which forces the existing approaches to resort to synthetic datasets such as the GoPro dataset [2]. 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 this, a real-world blur dataset for image deblurring should satisfy the following requirements as well. 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 image 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 4,554 pairs of blurred and ground truth sharp images of 234 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 mentioned above, we build an image acquisition system with two cameras and a beam splitter that can simultaneously capture a pair of blurred and sharp images that are geometrically aligned. We also develop a post-processing method to further align blurred and their corresponding

sharp images geometrically and photometrically as well as to remove noise.

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. Using our dataset, we benchmark existing deblurring methods and their limitations. 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 [11, 33, 6, 40, 41, 28, 35, 5, 22, 23] often model image blur using a convolution operation as:

$$b = k * l + n \tag{1}$$

Where b, l, and n denote a blurry image, a latent image, and additive noise, respectively. $*$ is a convolution operator. k is a blur kernel, which depicts the relative motion between the objects and the camera during exposure such as camera shakes and object motions. Based on this model, previous deblurring approaches solve an inverse problem to find both k and l from b. Unfortunately, such traditional approaches often fail to handle real-world blurred images because of the ill-posedness of the inverse problem, and their restrictive blur model. While several approaches have been proposed to address these issues by extending the blur model, their performance is still limited due to the increased ill-posedness of their models [16, 39, 7, 12, 15, 17].

To overcome such limitations of traditional approaches, several deep learning-based approaches have recently been proposed [26, 37, 19, 20]. In contrast to traditional approaches, they do not rely on any blur models, but directly learn a mapping function from a blurry image to its corresponding sharp image from a large collection of data. Consequently, they are not limited by blur models, and able to handle more realistic blurred images, e.g., with spatially-varying blurs caused by camera shakes and object motions. However, while they heavily rely on large-scale data to learn deblurring, there are no available datasets of real-world blurred images so far. Thus, all the existing approaches rely on synthetic datasets, which limit their performance on real-world blurred images.

Deblurring datasets. Along with deblurring methods, several datasets of blurred images have been proposed. However, most of them are designed not for learning but for evaluation of deblurring algorithms. Levin et al. [22] proposed a dataset of 32 images blurred by real camera shakes. To obtain real blurred images, they put four grayscale images on the wall, and captured them shaking a camera. Sun et al. [35] introduced a larger dataset

synthetically generated from 80 high-resolution natural images of diverse scenes and the eight blur kernels from Levin et al.'s dataset [22]. Köhler et al. [18] introduced a dataset of 48 blurred images with spatially-varying blur caused by real camera shakes. They also put four images on the wall, and captured them with 12 different camera shakes. All these datasets are too small to train neural networks, and not realistic enough as they are either synthetically generated or captured in controlled lab environments. Lai et al. [21] introduced a dataset of 100 real blurred images of various natural scenes for benchmarking different deblurring methods. However, their dataset is small as well, and does not provide ground truth sharp images, which are essential for learning image deblurring.

Recently, several synthetic datasets for image deblurring have been proposed [26, 25, 27, 34, 45]. To synthetically generate blurred images, they first captured sharp video frames using a high-speed camera, and blended 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 sharp frames, and because of the limited dynamic range of the sharp frames (Fig. 1). 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.

Figure 1. The synthetically blurred images in (a) and (c) in the GoPro dataset [26] are captured in well-lit environments and have unrealistic discontinuous blurs as shown in (b) and (d). Blending sharp images as done in the GoPro dataset cannot mimic saturated light streaks often observed in real-world blurred images like (e) due to the limited dynamic range of sharp images.

3. Image Acquisition System and Process

3.1 Image Acquisition System

To capture both blurred and sharp images simultaneously, we built a dual camera system shown in Fig. 2. Our system consists of a beam splitter and two mirrorless cameras so that the cameras can capture the same scene. The

cameras and the 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). They are synchronized by a multi-camera trigger to capture the same scene simultaneously. Our system is designed to use high-end mirrorless cameras with full-frame sensors and wide-angle lenses with short focal length based on the following reasons. First, we want to reflect the in-camera processing of conventional cameras into our dataset because it is much more frequent to encounter blurry JPEG images processed by camera ISPs than raw images. Second, full frame sensors and wide-angle lenses can gather a larger amount of light than smaller sensors and narrower-angle lenses so they can 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 [4, 13] 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.

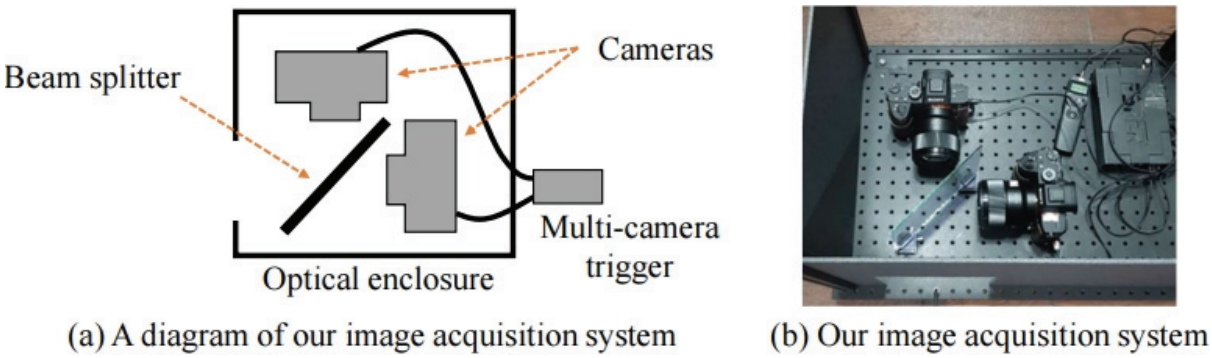


Figure 2. Our image acquisition system and its diagram

3.2 Image Acquisition Process

Using our image acquisition system, we captured blurred images of various scenes including both indoor and outdoor scenes. For each scene, we first captured a pair of two sharp images, which we refer to as a reference pair. 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 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. We captured 4,738 pairs of images of 234 different scenes including reference pairs. We captured all images both in the camera raw and JPEG formats, and generated two sets of datasets: *RealBlur-R* from the raw images, and *RealBlur-J* from the JPEG images processed by the camera ISP. Fig. 3 shows samples of blurred images in the *RealBlur* dataset.

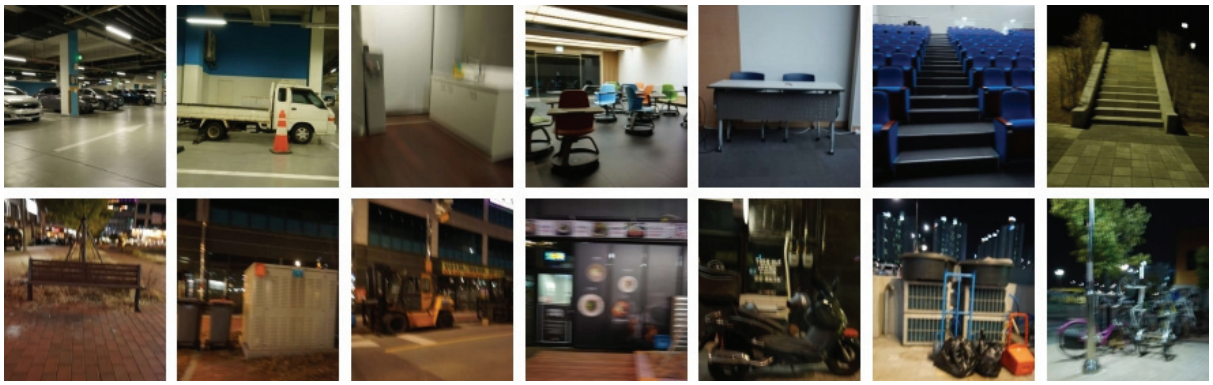


Figure 3. Blurred images in the *RealBlur* dataset. Our dataset consists of both dim-lit indoor and outdoor scenes where motion blur commonly occurs.

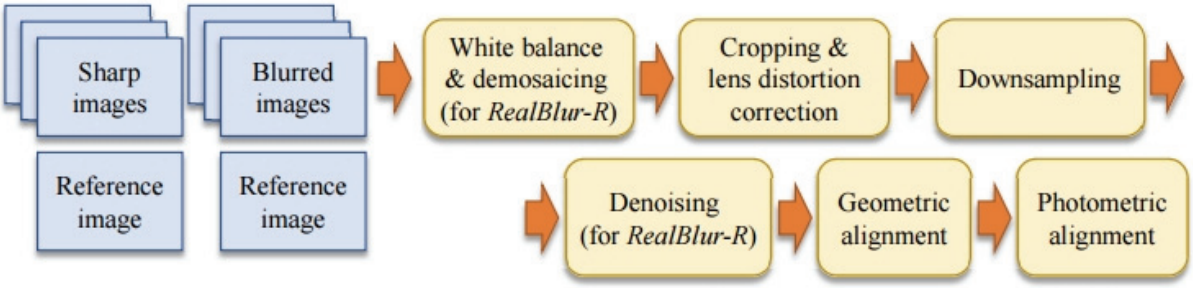


Figure 4. Overall procedure of our post-processing.

4. Post-Processing

The captured image pairs are post-processed for noise reduction, and geometric and photometric alignment. Fig. 4 shows an overview of our post-processing. We first briefly explain the post-processing procedure for *RealBlur-R*. For each pair of sharp and blurred images, we apply white balance and demosaicing to them in the first step. For white balance, we use the white balance parameters obtained from the cameras. For demosaicing, we use the adaptive homogeneity-directed demosaicing algorithm [14]. 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 invalid regions. We then correct lens

distortions from the cropped images to reduce the effect of lens distortion from geometric alignment in a later step. We estimate distortion parameters in a separate calibration step by capturing a checkerboard and using the method of Heikkila and Silven [13]. Then, we downsample the corrected images, and perform denoising to the downsampled sharp image. Finally, we perform geometric and photometric alignment. The size of original images from the cameras is 7952x5304. The image size after cropping is 2721x3094. The final image size after downsampling is 681x774.

For *RealBlur-J*, we follow the same procedure except for white balance, demosaicing, and denoising, as they are already 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, we use high ISO values to capture sharp images, which amplify noise in sharp images. As sharp images will be used as ground truth labels for training and evaluating deblurring methods, such amplified noise may adversely affect training and evaluation. 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 sharp images in the denoising step. For each sharp image, we estimate the amount of noise using Chen et al's method [4]. We then apply BM3D [9] to remove noise setting the noise level parameter to 1.5 times the estimated noise level.

We apply denoising only to 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

while capturing images due to camera shakes. To address this issue, for each scene, we compute a homography matrix from its reference pair. For homography estimation, we use an image registration method based on enhanced correlation coefficients [10] 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.

A homography estimated from a reference pair should align other image pairs of the same scene if the image pairs consist only of sharp images. However, as the other image pairs consist of blurred and sharp images, there still exist minuscule misalignment between blurred and sharp images due to their different exposure durations. Specifically, while the trigger synchronizes the shutters to open at the same time, they close at different moments as they have different exposure durations. As a result, simply using a homography of a reference pair aligns objects in a warped sharp image to one corner of their corresponding blurry objects, not the center (Fig. 5(b)).

Traditional blur model-based deblurring approaches such as [32, 3, 8] often align images or blur kernels with respect to the centers of mass, or the 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. We will show in Sec. 5. 1 that this seemingly simple alignment scheme that matches the centers of blurry and sharp objects is actually the most effective for learning of deblurring.

The additional geometric alignment step consists of two steps for robust alignment. In the first step, we compute a similarity transform using phase correlation [30], which we found robust to camera shakes (see appendix for more details). 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.

In the second step, we further align images using translational shifts to accurately match the centers of objects. 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 [6, 41, 5], we estimate a blur kernel k by minimizing the following energy function:

$$E(k) = \mathrm{argmin}_k[||k * \nabla s - \nabla b||^2 + \lambda||\nabla k||^2] \quad (2)$$

where s and b are sharp and blurred images, respectively, and ∇ is a gradient operator. λ is a weight for the regularization term, 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)). The effect of the centroid-based alignment will be discussed in Sec. 5.

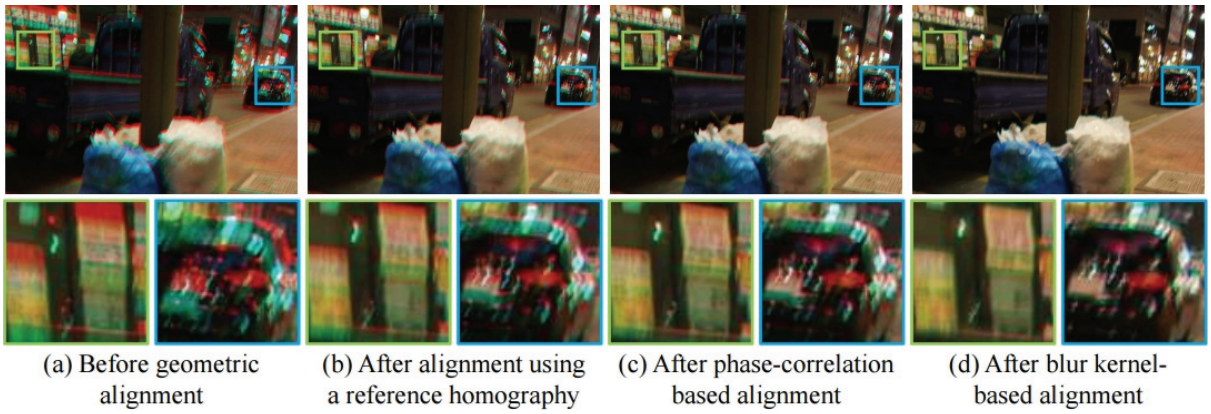


Figure 5. Our geometric alignment. Each image shows an alignment result as a stereo-anaglyph image, where the sharp and blurred images are visualized using red and cyan, respectively, and overlaid to each other. The blurred image has slightly non-uniform blur due to camera shakes so the shapes of cyan light streaks differ across different regions.

4.3 Photometric Alignment

Although we use cameras and lenses of the same models, there may exist slight difference of intensities between their images. To resolve this, similarly to [2, 29, 1], we adopt a linear model for photometric alignment, which is defined as:

$$r_1 = \alpha r_2 + \beta \tag{3}$$

where r_1 and r_2 are images of a reference pair after geometric alignment. α and β are coefficients of the linear model. We estimate α and β as $\alpha = \sigma_1/\sigma_2$ and $\beta = \mu_1 - \alpha\mu_2$ where σ_1 and σ_2 are the standard deviations of r_1 and r_2, respectively, and μ_1 and μ_2 are the means of r_1 and r_2, respectively. Finally, we transform sharp images with α and β to match the intensities of blurred images. To process color images, we apply the linear 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 investigate the effect of our dataset. All data including the *RealBlur* dataset will be available on our project webpage.

Datasets and evaluation measure. For the benchmark, we randomly select 184 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 184 reference pairs, and 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 [26], and K*ö*hler *et al.*'s [18]. 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. K*ö*hler *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 [24] as follows. For each image in BSD500, we randomly generate 40 synthetic uniform motion blur kernels using Schmidt *et al.*'s method [31] 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 results with translational shifts even when blur is successfully removed. To deal with such shifts when measuring the quality of deblurring results against their corresponding ground truth images, we follow K*ö*hler *et al.*'s approach [18] in all the experiments in this section, which first aligns a deblurring result to its ground truth sharp image using phase correlation, and computes PSNRs or SSIMs [38].

5.1 Analysis of RealBlur Dataset

Blur size distribution. To show the coverage of our dataset in the aspect of blur, we report the blur sizes in our dataset. For analysis, we estimated blur kernels for all images and manually inspected their sizes. Fig. 6(a) shows the distribution of the blur sizes in our dataset. The distribution shows a skewed and heavy-tailed shape, where most of the blurs are relatively small having sizes of around 5 to 20 pixels, while there exist significantly large blurs up to 70 pixels, which is 10% of the width of an image (Fig. 6 (b)). This is because every blurred image has a certain amount of blur due to the long exposure time of 1/2 sec., while large blurs occur less frequently.

Noise. Table 1 reports the average noise levels of the images in our dataset, which are estimated by Chen *et al*'s method [4]. The table also reports the noise levels of the GoPro [26] and Reds [46] datasets, which are the most widely used deblurring datasets, for reference. Both datasets consist of denoised images by camera ISPs. As we use high ISO values, the sharp images in *RealBlur-R* before denoising have a relatively higher level of noise. After denoising, the sharp images in *Realblur-R* have similar noise levels compared to the GoPro [2] and Reds [46]

datasets. The sharp and blurred images in the *RealBlur-J* dataset also show similar noise levels to the other denoised images as the dataset consists of images processed by camera ISPs.

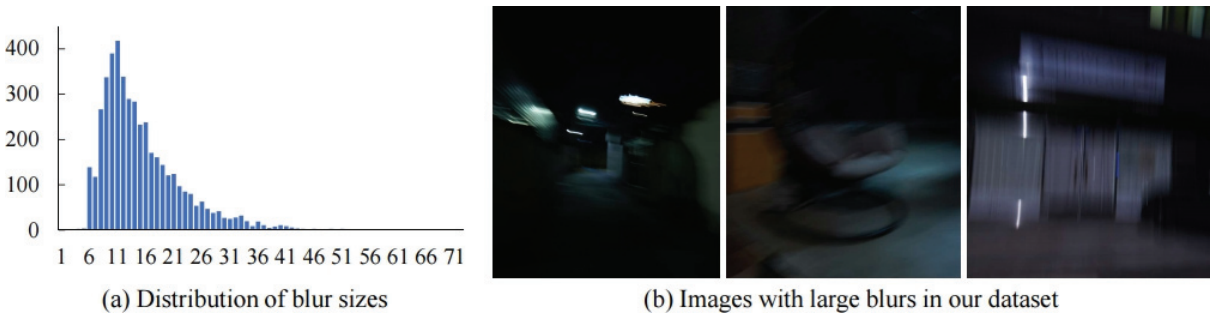


Figure 6. The distribution of blur sizes in the *RealBlur* dataset. The dataset mostly covers small-scale blurs while it also includes extremely large blurs shown in (b).

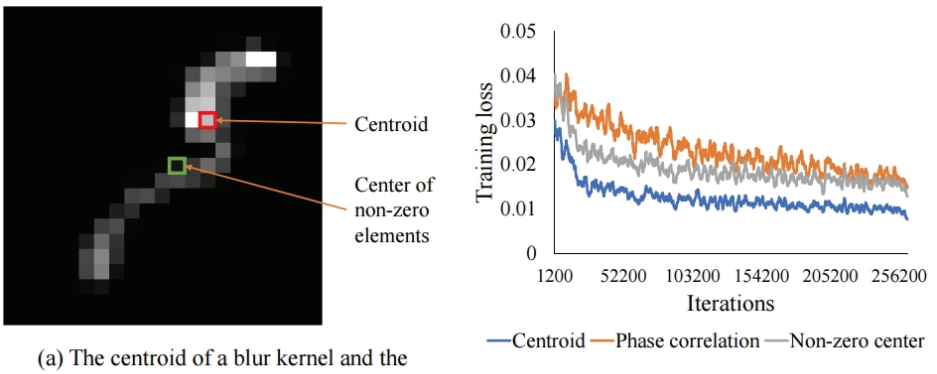
Table 1. Noise levels of the images in the *RealBlur* dataset in terms of standard deviation of Gaussian noise.

Dataset		**Noise level**
RealBlur-R	Sharp images before denoising	2.9731
	Sharp images after denoising	0.1543
	Blurred images	0.6642
RealBlur-J	Sharp images	0.2219
	Blurred images	0.3742
GoPro [26]	Sharp images	0.3474
	Blurred images	0.3042
Reds [46]	Sharp images	0.357
	Blurred images	0.3868

Effect of geometric alignment. We first analyze the effect of geometric alignment with respect to the centroids of blur kernels on learning of image deblurring. Restoring a sharp pixel requires information from nearby pixels in a blurred image, which sets a lower bound of the receptive field size required for deblurring. Alignment with respect to 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 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 the center of the non-zero elements is closest to all non-zero elements (Fig. 6(a)). We refer to this center as the non-zero center for brevity. However, we found that this approach is less effective than using the centroids as shown below. While it is unclear why, we conjecture that it is because the centroid is the most central position in terms of the amount of information 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 the BSD-B dataset: aligned using translation estimated by the phase correlation [30], aligned to the centers of the non-zero elements of the blur kernels, and aligned to the centroids of the blur kernels. We train SRN-DeblurNet of Tao *et al.* [37], which is a state-of-the-art deep learning-based approach, with the three sets separately, and compare their performance on Köhler *et al.*'s dataset [18]. The average PSNR values of the phase correlation-, non-zero center-, and centroid-based datasets are 26.49, 27.53, and 28.10 dBs, respectively. Moreover, the centroid-based alignment also results in the most stable and efficient training as shown in Fig. 6(b). This result proves that the alignment based on the centroids of blur kernels in our post-procesing is essential for effective learning of image deblurring.



(a) The centroid of a blur kernel and the center of its non-zero elements

(b) Training losses of different alignment schemes

Figure 7. Training losses of different alignment methods.

Geometric alignment accuracy. Our geometric alignment assumes a couple of assumptions. First, a blurred image and its corresponding sharp image 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 pair of blurred and sharp images as follows. For each pair of blurred and sharp images that are aligned, we first divide them into a 2x2 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 2 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 different exposure times of the blurred and sharp images (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 2x2 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.

Table 2. Displacement error of variants of our geometric alignment process.

Geometric alignment methods	Error (pixels)
Reference homography	4.9934
Reference homography+blur kernel	0.7518
Reference homography+phase correlation+blur kernel	0.6981

5.2 Benchmark

Dataset comparison and training strategy. Before benchmarking existing deblurring approaches, we first compare the performance of our datasets with 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 [37]. 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 appendix for

details about training.

Tables 3 and 4 show the performance of different combinations of the training sets on different test sets in terms of PSNR and SSIM [38]. 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 3 and 4) achieves lower performance on our real-world blur test sets compared to the other combinations in general, which proves that the GoPro dataset is not realistic enough to cover real-world low-light images. The BSD-B dataset (2nd rows in Tables 3 and 4) also achieves low performance on the real blur test sets, but high performance on K*ö*hler *et al.*'s test set, which is possibly because K*ö*hler *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 3 and 4) achieve higher performance on the real blur test sets, which validates the necessity of real training data for deblurring of real-world blurred images.

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.

Table 3. Performance comparison of different datasets. We trained a deblurring network of Tao et al. [37] using different training sets and measured its performance.

Training Sets				**Test Sets (PSNR/SSIM)**		
RealBlur-R	**GoPro**	**BSD-B**	**Pre-trained**	***RealBlur-R***	**Köhler**	**GoPro**
	V			36.04/0.9526	26.78/0.7960	30.29/0.9350
		V		35.65/0.9282	28.10/0.8281	27.17/0.8868
V				37.09/0.9551	24.67/0.7391	23.25/0.8354
V	V			37.33/0.9565	25.54/0.7584	27.36/0.8896
V		V		38.83/0.9663	27.85/0.8212	27.20/0.8878
V	V	V		38.91/0.9666	27.88/0.8220	29.41/0.9234
V			V	38.83/0.9660	26.01/0.7817	25.37/0.8585
V	V		V	38.93/0.9667	26.95/0.7997	30.02/0.9317
V		V	V	39.04/0.9676	28.10/0.8274	27.15/0.8849
V	V	V	V	38.94/0.9674	28.02/0.8283	29.71/0.9261

Table 4. Performance comparison of different datasets. We trained a deblurring network of Tao et al. [37] using different training sets and measured its performance.

Training Sets				Test Sets (PSNR/SSIM)		
RealBlur-J	**GoPro**	**BSD-B**	**Pre-trained**	***RealBlur-J***	**Köhler**	**GoPro**
	V			28.73/0.8829	26.78/0.7960	30.29/0.9350
		V		28.97/0.8853	28.10/0.8281	27.17/0.8868
V				31.25/0.9124	26.29/0.7886	25.29/0.8569
V	V			31.37/0.9150	26.67/0.7988	29.25/0.9211
V		V		31.48/0.9184	27.74/0.8225	27.12/0.8868
V	V	V		31.65/0.9191	27.41/0.8178	29.37/0.9229
V			V	31.49/0.9195	26.67/0.8004	25.61/0.8645
V	V		V	31.54/0.9203	26.93/0.8048	29.97/0.9310
V		V	V	31.55/0.9224	27.89/0.8283	27.19/0.8866
V	V	V	V	31.52/0.9217	27.58/0.8223	29.72/0.9266

Benchmarking deblurring methods. We then benchmark state-of-the-art deblurring methods including both traditional optimization-based and recent deep learning-based approaches using our test sets. Regarding traditional approaches, we include the methods of Xu *et al.* [41] and Pan *et al.* [28]. Regarding deep learning-based approaches, we include the methods of Nah *et al.* [26], Zhang *et al.* [43], Zhang *et al.* [42], DeblurGAN [19], DeblurGAN-v2 [20], and SRN-DeblurNet [37]. For all the deep learning-based approaches, we use pre-trained models provided by the authors. For the models of DeblurGAN-v2 [20] and SRN-DeblurNet [36], we also include models trained with our training strategy.

Table 5 shows a summary of the benchmark. In the benchmark, the optimization-based 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 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. 8 shows a qualitative comparison of the deep learning-based methods in Table 5. The blurred image is captured with camera shakes outside at night. All the methods trained without real-world blurred images fail to

restore light streaks as well as other image details. On the other hand, the methods trained with our datasets show better restored results. The results of SRN-DeblurNet [37] trained with our dataset in Fig. 8(i) and (s) show accurately restored image details and lights. The results of DeblurGAN-v2 [20] trained with our dataset in Fig. 8(g) and (q) also show better restored details than the others, while slightly worse than those of SRN-DeblurNet trained with our datasets.

Table 5. 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.

RealBlur-J		***RealBlur-R***	
Methods	**PSNR/SSIM**	**Methods**	**PSNR/SSIM**
SRN-DeblurNet* [37]	31.52/0.9217	SRN-DeblurNet* [37]	38.94/0.9674
DeblurGAN-v2* [20]	30.19/0.8957	DeblurGAN-v2* [20]	37.12/0.9520
DeblurGAN-v2 [20]	28.95/0.8831	Zhang *et al.* [43]	36.16/0.9539
Zhang *et al.* [43]	28.78/0.8805	SRN-DeblurNet [37]	36.04/0.9526
SRN-DeblurNet [37]	28.73/0.8829	Zhang *et al.* [42]	35.89/0.9507
Nah *et al.* [26]	28.43/0.8551	DeblurGAN-v2 [20]	35.84/0.9505
DeblurGAN [19]	28.35/0.8603	Xu *et al.* [41]	34.79/0.9412
Zhang *et al.* [42]	28.07/0.8667	Nah *et al.* [26]	34.31/0.8886
Xu *et al.* [41]	27.15/0.8501	Pan *et al.* [28]	34.23/0.9219
Pan *et al.* [28]	27.10/0.8127	DeblurGAN [19]	33.88/0.9077

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. 9 shows results of deep learning-based methods with different training sets. The blurred images in Fig. 9 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. 9(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 a large number of dynamic scenes.

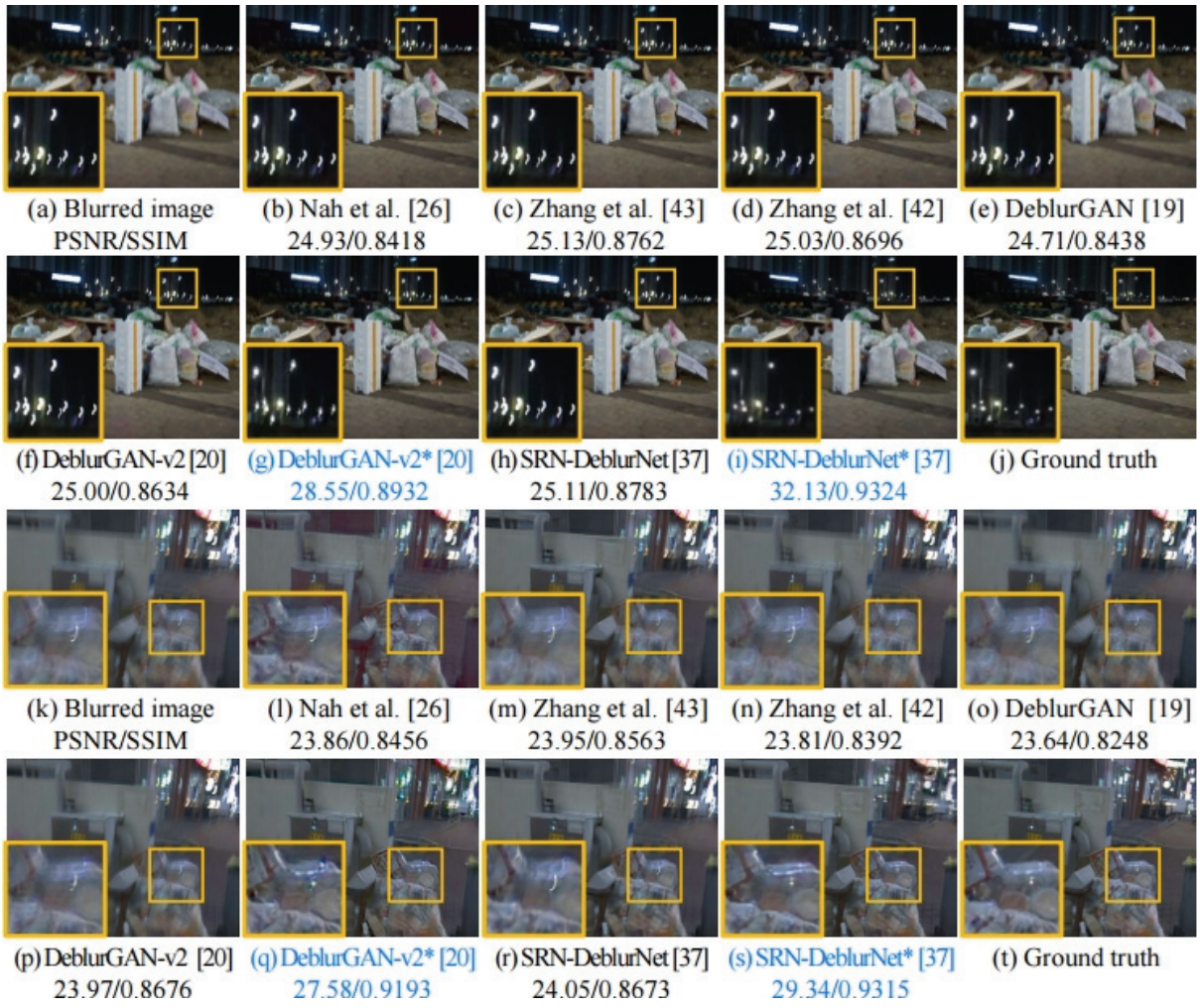


Figure 8. Qualitative comparison of different deblurring methods on the *RealBlur-J* and *RealBlur-R* test sets. (a)-(j) Deblurring results using *RealBlur-J*. (k)-(t) Deblurring results using *RealBlur-R*. For visualization, the examples of *RealBlur-R* are gamma-corrected. Methods marked by `*' in blue are trained with our datasets.

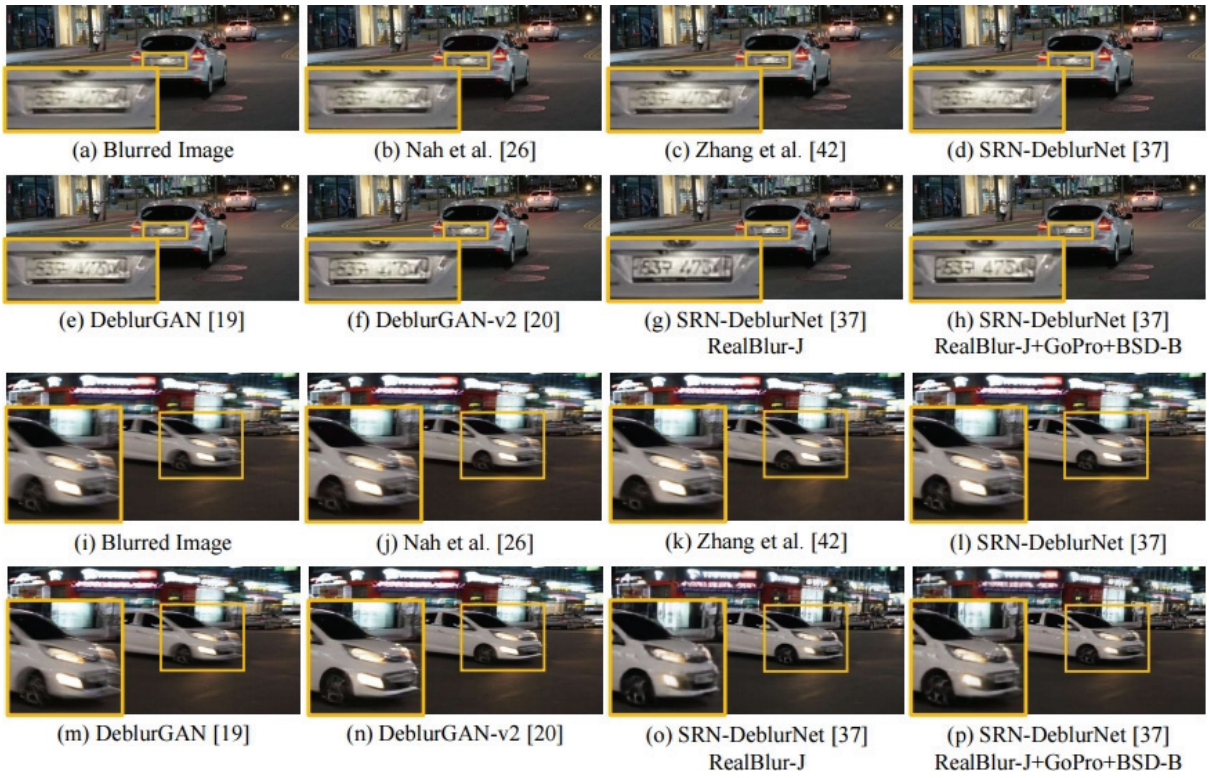


Figure 9. Qualitative comparison of different methods on images of dynamic scenes.

Generalization to Different Lenses. We present an additional experimental result that shows the generalization ability of deblurring networks trained with our dataset. Telephoto lenses produce blurry images with different characteristics from wide-angle lenses used in our system. They often cause both camera shake and defocus blur particularly easily due to their narrow fields of view and shallow depths of field. Thus, in this experiment, we study how deblurring networks trained with our dataset that does not have defocus blur generalize to blurry images taken using telephoto lenses.

To this end, we created additional test sets of real-world blurred images taken using telephoto lenses. To create the test sets, we replaced the lenses in our image acquisition system with telephoto lenses (SEL85F18), and captured 996 pairs of images from 50 different scenes. We then applied our postprocessing to the images with some minor changes. Specifically, we downsampled the images by 1/8 as blurs caused by telephoto lenses tend to be larger. We also omitted cropping and lens distortion correction as the images taken using telephoto lenses have neither invalid regions nor noticeable lens distortion. Finally, we obtained two additional test sets: *RealBlur-TeleR* from raw images and *RealBlur-TeleJ* from JPEG images. The resulting blurred images in the test sets have both motion and defocus blurs, while the sharp images have only defocus blurs.

We then evaluate the performance of deep learning-based deblurring methods on *RealBlur-TeleR* and *RealBlur-*

TeleJ. For the evaluation on *RealBlur-TeleJ*, we trained DeblurGAN-v2 [20] and SRN-DeblurNet [37] with *RealBlur-J* as well as the GoPro [26] and BSD-B datasets as described in the paper. For the evaluation on *RealBlur-TeleR*, we trained the networks with *RealBlur-R* instead of *RealBlur-J*.

Fig. 10 shows a qualitative comparison. In the figure, the blurred images on the top and on the bottom are from *RealBlur-TeleJ* and *RealBlur-TeleR*, respectively. The blurred images in the figure have both camera shakes and defocus blur. While the other networks trained with only synthetic datasets fail to restore sharp images, the networks trained with our datasets (marked in blue with `*') successfully remove camera shakes. Regarding defocus blur, interestingly, the networks trained with our dataset keep defocus blur intact while removing motion blur even though they are trained using images without defocus blur. It will be interesting future work to investigate how networks remove motion blur selectively, and how to handle both defocus and motion blur properly. Table 6 shows a quantitative comparison. The table shows that the networks trained with *RealBlur-R* and *RealBlur-J* clearly outperform the other networks even on *RealBlur-TeleR* and *RealBlur-TeleJ* achieving high PSNR and SSIM values.

Table 6. Benchmark of deep learning-based deblurring methods on *RealBlur-TeleJ* and *RealBlur-TeleR*. Blue*: networks trained with *RealBlur*. Black: networks trained only with synthetic datasets. The methods are sorted in the descending order with respect to PSNR.

RealBlur-TeleJ		***RealBlur-TeleR***	
Methods	**PSNR/SSIM**	**Methods**	**PSNR/SSIM**
SRN-DeblurNet* [37]	29.94/0.8993	SRN-DeblurNet* [37]	38.74/0.9577
DeblurGAN-v2* [20]	29.14/0.8756	DeblurGAN-v2* [20]	37.27/0.9446
DeblurGAN-v2 [20]	27.79/0.8610	SRN-DeblurNet [37]	36.55/0.9437
SRN-DeblurNet [37]	27.22/0.8496	Zhang *et al.* [43]	36.53/0.9414
Zhang *et al.* [43]	27.08/0.8420	DeblurGAN-v2 [20]	36.27/0.9402
DeblurGAN [19]	26.91/0.8291	DeblurGAN [19]	33.26/0.8859
Nah *et al.* [26]	26.84/0.8185	Zhang *et al.* [43]	36.13/0.9377
Zhang *et al.* [42]	26.33/0.8225	Nah *et al.* [26]	34.95/0.8898

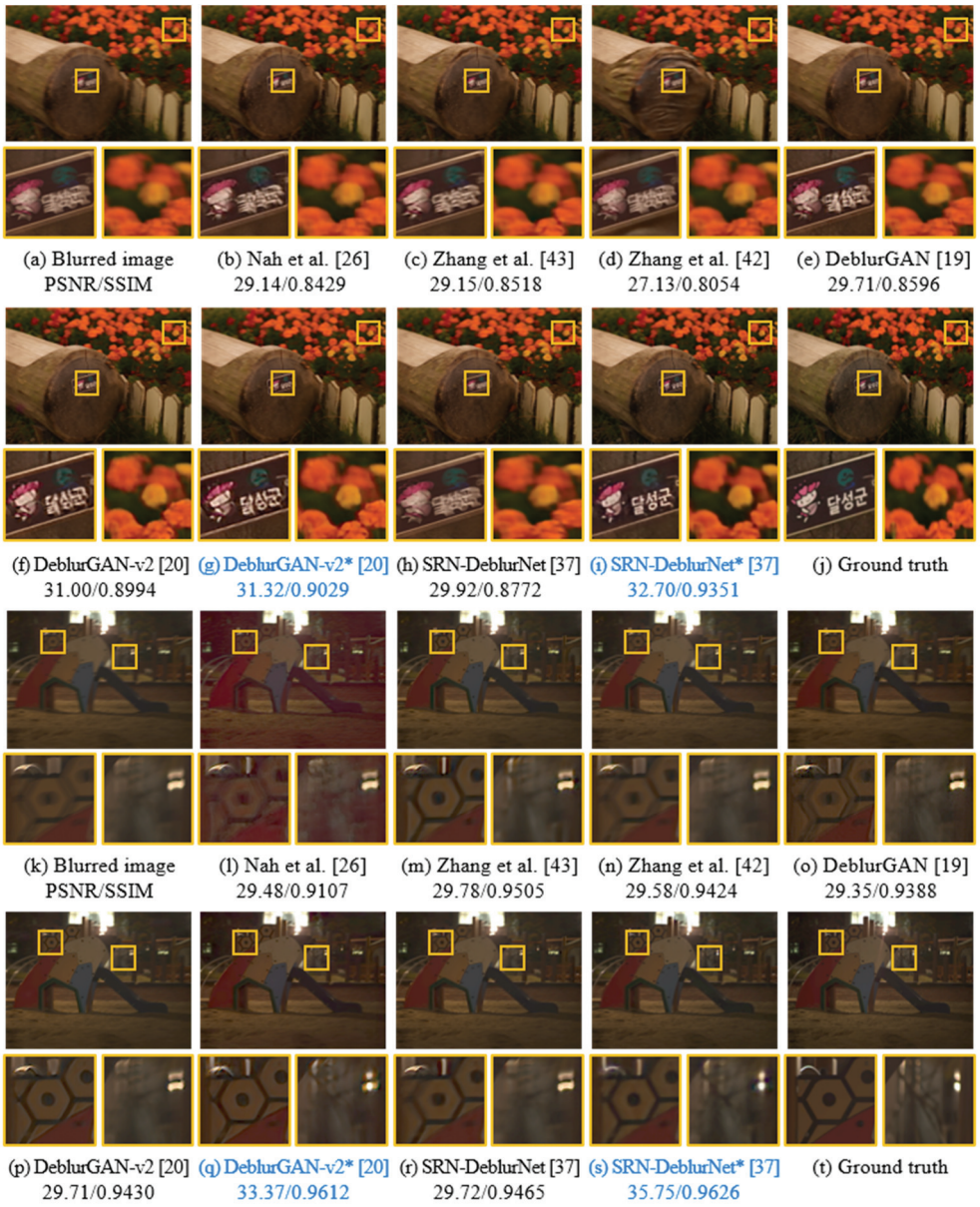


Figure 10. Qualitative comparison on *RealBlur-TeleJ* and *RealBlur-TeleR*. The networks marked in blue with `*' are trained with our dataset. The blurred image in (a) is from *RealBlur-TeleJ*. DeblurGAN-v2 [20] and SRN-DeblurNet [37] in (g) and (i) are trained with *RealBlur-J*. The blurred image in (k) is from *RealBlur-TeleR*. DeblurGAN-v2 and SRN-DeblurNet in (q) and (s) are trained with *RealBlur-R*. Below each result, its PSNR and SSIM values are reported.

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 post-processing process for geometric and photometric alignment as well as noise removal, and analyzed the effect and accuracy of our 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.

7. Appendix

Phase correlation in the presence of blur. Our postprocessing uses the phase correlation-based image registration method of Reddy and Chatterji [30] for aligning a pair of sharp and blurred images as we found it robust to blur in practice. In this section, we provide a theoretical background of the robustness of the phase correlation-based alignment to blur.

We first consider a simple case for image alignment where we have two images f_1 and f_2 that have the same content with a slight translational shift. Then, we can define the relationship between them as:

$$\mathrm{f}_2(\mathrm{x},\mathrm{y}) = \mathrm{f}_1(\mathrm{x}-\mathrm{x}_0,\mathrm{y}-\mathrm{y}_0) \tag{4}$$

where x and y are pixel coordinates, and x_0 and y_0 are translational shifts along the x and y axes, respectively. The phase correlation-based alignment proposed by Reddy and Chatterji [30] uses the Fourier domain to find x_0 and y_0. Applying the Fourier transform to Eq. (4), we can obtain:

$$\mathrm{F}_2(\mathrm{u},\mathrm{v}) = \mathrm{F}_1(\mathrm{u},\mathrm{v})\exp\big(-\mathrm{i}(\mathrm{ux}_0+\mathrm{vy}_0)\big) \tag{5}$$

where F_1 and F_2 are the Fourier transforms of f_1 and f_2, respectively. u and v are 2-dimensional indices in the Fourier domain. i is the imaginary unit defined as $\mathrm{i}^2 = -1$.

Each component of a Fourier transform can be decomposed into its amplitude and phase components as:

$$\mathrm{F(u,v)} = |\mathrm{F(u,v)}| \exp\big(-\mathrm{i}\phi(\mathrm{u,v})\big) \tag{6}$$

where $F(u,v)$ and $\phi(u,v)$ are the amplitude and phase of $F(u,v)$, respectively. Based on this, Reddy and Chatterji define a cross-power spectrum S of the images f_1 and f_2 to extract the phase difference between them as follows:

$$S(u,v) = \frac{\mathrm{F_2(u,v)F_1^*(u,v)}}{|\mathrm{F_2(u,v)F_1^*(u,v)}|} = \exp\big(-\mathrm{i(ux_0 + vy_0)}\big) \tag{7}$$

where $F_1^*(u,v)$ is the complex conjugate of $F_1(u,v)$. By taking inverse Fourier transform of $S(u,v)$, we can obtain an impulse function that is approximately zero everywhere except at (x_0, y_0). Thus, we can locate the maximum component from the inverse Fourier transform of S to find the optimal parameters for image alignment. Reddy and Chatterji [7] extend this approach to similarity transforms.

We can extend the approach described above for a pair of blurred and sharp images. We assume that a blurred image g and its corresponding sharp image f has the relationship defined as:

$$\mathrm{g(x,y)} = (\mathrm{f} * \mathrm{h})(\mathrm{x,y}) \tag{8}$$

where h is a blur kernel or a point spread function, and $*$ is the convolution operator. Eq. (8) is a generalization of Eq. (4) as h in Eq. (8) can describe translational shift between g and f. By applying the Fourier transform to Eq. (8), we can obtain:

$$\mathrm{G(u,v)} = \mathrm{F(u,v)H(u,v)} \tag{9}$$

where G, F, and H are Fourier transforms of g, f, and h, respectively. The cross-power spectrum of G and F is then defined as:

$$\mathrm{S(u,v)} = \frac{\mathrm{G(u,v)F^*(u,v)}}{|\mathrm{G(u,v)F^*(u,v)}|} = \frac{\mathrm{H(u,v)F(u,v)F^*(u,v)}}{|\mathrm{H(u,v)F(u,v)F^*(u,v)}|} = \frac{\mathrm{H(u,v)}}{|\mathrm{H(u,v)}|} \tag{10}$$

where $\frac{H(u,v)}{|H(u,v)|}$ is the phase component of the blur kernel h. The inverse Fourier transform of $S(u,v)$ is no longer an impulse function, but the phase-only reconstruction of the blur kernel h, which still looks similar to h as shown in Fig. 10. Locating the maximum component of the phase-only reconstruction of h aligns f to a certain position of b, although it does not guarantee to align the objects in f to the centers of blurry objects in g.

Unfortunately, the argument above does not generalize to similarity transforms as shown in [47] or to spatially-varying blur. Nonetheless, thanks to our physically-aligned imaging acquisition system, we found that the phase correlation-based alignment works well in general on our dataset increasing the overall alignment accuracy between the sharp and blurred images as reported in our main paper.

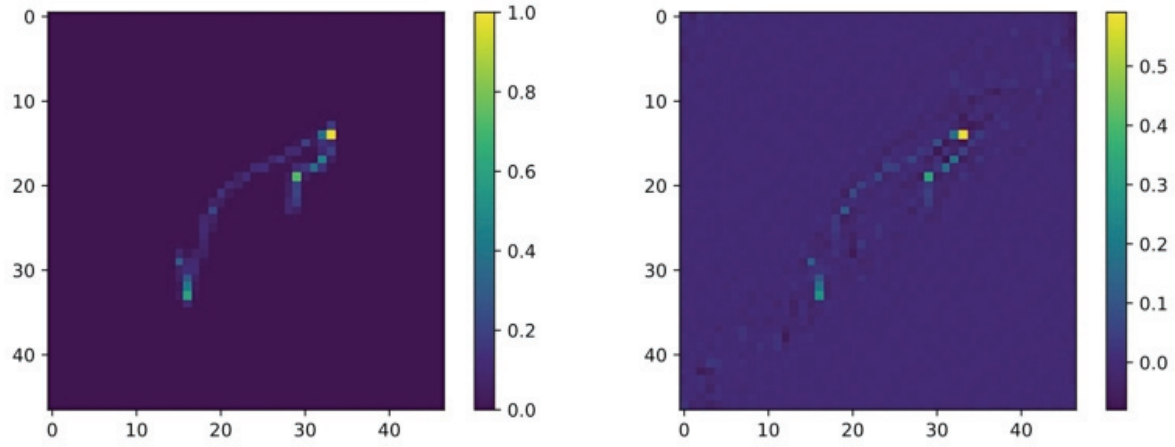


Figure 11. Phase-only reconstruction of a blur kernel

Training Details. We use the source code of the authors provided on their project webpage for training SRN-DeblurNet [37]. Specifically, we use the color version of SRN-DeblurNet without LSTM as suggested by the authors. We train the network with its default training setting except the number of iterations, which is set to 262,000 for training both from scratch and from a pre-trained model in our experiments. For training from a pre-trained model, we use the pre-trained model of the authors. For training from scratch, we initialize the network as described in [37]. For training the network using all three datasets, which are ours, BSD-B and GoPro [26], we oversample our dataset six times, and the GoPro dataset 10 times to balance the datasets of different sizes.

We use the framework with Inception-ResNet-v2 as its backbone, and the double-scale discriminator with the RaGAN-LS loss [20] for training DeblurGAN-v2 [20]. We use the source code provided by the authors. We train

the network using the Adam optimizer with hyperparameters described in [20] except the number of iterations. Specifically, we train the network with the learning rate of 10^{-4} for 150,000 iterations, then linearly decay the learning rate to 10^{-7} for another 150,000 iterations. For training the network with our dataset and the BSD-B and GoPro [26] datasets, we oversample our dataset six times, and the GoPro dataset 10 times to balance the datasets of different sizes.

8. References

[1]. Abdelhamed, A., Lin, S., Brown, M.S.: A high-quality denoising dataset for smart phone cameras. In: CVPR (June 2018)

[2]. 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)

[3]. 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)

[4]. Chen, G., Zhu, F., Ann Heng, P.: An efficient statistical method for image noise level estimation. In: ICCV (December 2015)

[5]. Cho, S., Lee, S.: Convergence analysis of map-based blur kernel estimation. In: ICCV. pp. 4818–4826 (Oct 2017)

[6]. Cho, S., Lee, S.: Fast motion deblurring. ACM Trans. Graph. 28(5), 145:1–145:8 (Dec 2009)

[7]. Cho, S., Wang, J., Lee, S.: Handling outliers in non-blind image deconvolution. In: ICCV (2011)

[8]. Cho, T.S., Paris, S., Horn, B.K.P., Freeman, W.T.: Blur kernel estimation using the radon transform. In: CVPR (2011)

[9]. 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)

[10]. Evangelidis, G.D., Psarakis, E.Z.: Parametric image alignment using enhanced correlation coefficient maximization. TPAMI 30(10), 1858–1865 (Oct 2008)

[11]. 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)

[12]. Gupta, A., Joshi, N., Zitnick, L., Cohen, M., Curless, B.: Single image deblurring using motion density functions. In: ECCV. pp. 171–184 (2010)

[13]. 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)

[14]. Hirakawa, K., Parks, T.W.: Adaptive homogeneity-directed demosaicing algorithm. TIP 14(3), 360–369 (March 2005)

[15]. Hirsch, M., Schuler, C.J., Harmeling, S., Schölkopf, B.: Fast removal of non-uniform camera shake. In: ICCV. pp. 463–470 (2011)

[16]. Hu, Z., Cho, S., Wang, J., Yang, M.H.: Deblurring low-light images with light streaks. In: CVPR. pp. 3382–3389 (2014)

[17]. Kim, T.H., Lee, K.M.: Segmentation-free dynamic scene deblurring. In: CVPR. pp. 2766–2773 (June 2014)

[18]. 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)

[19]. Kupyn, O., Budzan, V., Mykhailych, M., Mishkin, D., Matas, J.: DeblurGAN: Blind motion deblurring using conditional adversarial networks. In: CVPR (June 2018)

[20]. Kupyn, O., Martyniuk, T., Wu, J., Wang, Z.: DeblurGAN-v2: Deblurring (orders-of-magnitude) faster and better. In: ICCV (October 2019)

[21]. 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)

[22]. Levin, A., Weiss, Y., Durand, F., Freeman, W.T.: Understanding and evaluating blind deconvolution algorithms. In: CVPR. pp. 1964–1971 (2009)

[23]. Levin, A., Weiss, Y., Durand, F., Freeman, W.T.: Efficient marginal likelihood optimization in blind deconvolution. In: CVPR. pp. 2657–2664 (2011)

[24]. 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)

[25]. 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) Work shops (June 2019)

[26]. Nah, S., Hyun Kim, T., Mu Lee, K.: Deep multi-scale convolutional neural network for dynamic scene deblurring. In: CVPR (July 2017)

[27]. 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)

[28]. Pan, J., Sun, D., Pfister, H., Yang, M.H.: Blind image deblurring using dark channel prior. In: CVPR. pp. 1628–1636 (2016)

[29]. Plotz, T., Roth, S.: Benchmarking denoising algorithms with real photographs. In: CVPR (July 2017)

[30]. 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)

[31]. 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

[32]. Schuler, C.J., Hirsch, M., Harmeling, S., Schölkopf, B.: Learning to deblur. TPAMI 38(7), 1439–1451 (July 2016)

[33]. 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)

[34]. 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)

[35]. Sun, L., Cho, S., Wang, J., Hays, J.: Edge-based blur kernel estimation using patch priors. In: ICCP (2013)

[36]. Tai, Y.W., Du, H., Brown, M.S., Lin, S.: Image/video deblurring using a hybrid camera. In: CVPR (2008)

[37]. Tao, X., Gao, H., Shen, X., Wang, J., Jia, J.: Scale-recurrent network for deep image deblurring. In: CVPR (June 2018)

[38]. 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

[39]. Whyte, O., Sivic, J., Zisserman, A., Ponce, J.: Non-uniform deblurring for shaken images. In: CVPR. pp. 491–498 (2010)

[40]. Xu, L., Jia, J.: Two-phase kernel estimation for robust motion deblurring. In: ECCV (2010)

[41]. Xu, L., Zheng, S., Jia, J.: Unnatural L0 sparse representation for natural image deblurring. In: CVPR (2013)

[42]. Zhang, H., Dai, Y., Li, H., Koniusz, P.: Deep stacked hierarchical multi-patch network for image deblurring. In: CVPR (June 2019)

[43]. 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)

[44]. Zhang, Z.: A flexible new technique for camera calibration. TPAMI 22(11), 1330–1334 (Nov 2000)

[45]. 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)

[46]. 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) Work-shops (June 2019)

[47]. Ojansivu, V., Heikkilä, J.: Blur invariant registration of rotated, scaled and shifted images. In: 2007 15th European Signal Processing Conference. pp. 1755–1759 (Sep 2007)

요 약 문

Real-World Blur Dataset for Learning and Benchmarking Deblurring Algorithms

최근의 디블러링 연구들은 딥러닝을 활용하여 블러 된 영상으로부터 선명한 영상을 복원하도록 학습한다. 디블러링의 경우 실제 블러 된 영상과 그에 대응하는 선명한 영상을 얻기 힘들어 데이터셋이 존재하지 않아 기존의 방법들은 학습을 위해 선명한 영상으로부터 가상으로 블러 된 영상을 생성하여 학습에 활용하는 한계가 있다. 본 논문에서는 빔 스플리터를 활용해 블러 영상과 선명한 영상을 동시에 촬영할 수 있는 카메라 시스템을 제작하여 실제 블러 영상 데이터셋을 수집하였다. 또한 보다 정확한 정답 영상을 얻기 위해 선명한 영상과 블러된 영상에서의 영상 정합(Geometric alignment) 및 색 조정(Photometric alignment) 방법을 제안하였다. 수집한 데이터셋을 통해 기존의 딥러닝 기반 방법들이 실제 영상에서 제대로 동작하지 않음을 보이고 실제 영상으로 학습을 진행함으로써 보다 강건함을 보였다.

핵심어: 디블러링, 영상 복원, 데이터셋, 영상 정합