

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

## — Supplementary Material I —

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

### 1 Phase Correlation in the Presence of Blur

Our postprocessing uses the phase correlation-based image registration method of Reddy and Chatterji [8] 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:

$$f_2(x, y) = f_1(x - x_0, y - y_0) \quad (1)$$

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 [8] uses the Fourier domain to find  $x_0$  and  $y_0$ . Applying the Fourier transform to Eq. (1), we can obtain:

$$F_2(u, v) = F_1(u, v) \exp(-i(ux_0 + vy_0)) \quad (2)$$

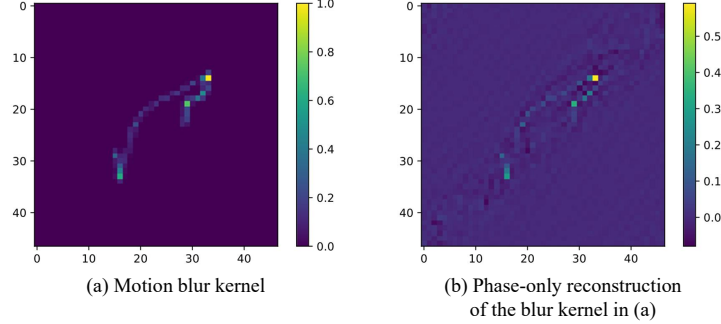
where  $F_1$  and  $F_2$  are the Fourier transforms of  $f_1$  and  $f_2$ , respectively.  $u$  and  $v$  are 2-dimensional indices in the Fourier domain.  $i$  is the imaginary unit defined as  $i^2 = -1$ .

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

$$F(u, v) = |F(u, v)| \exp(-i\phi(u, v)) \quad (3)$$

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:

$$S(u, v) = \frac{F_2(u, v)F_1^*(u, v)}{|F_2(u, v)F_1^*(u, v)|} = \exp(-i(ux_0 + vy_0)) \quad (4)$$



**Fig. 1.** Phase-only reconstruction of a blur kernel.

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 [8] 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:

$$g(x, y) = (f * h)(x, y) \quad (5)$$

where  $h$  is a blur kernel or a point spread function, and  $*$  is the convolution operator. Eq. (5) is a generalization of Eq. (1) as  $h$  in Eq. (5) can describe translational shift between  $g$  and  $f$ . By applying the Fourier transform to Eq. (5), we can obtain:

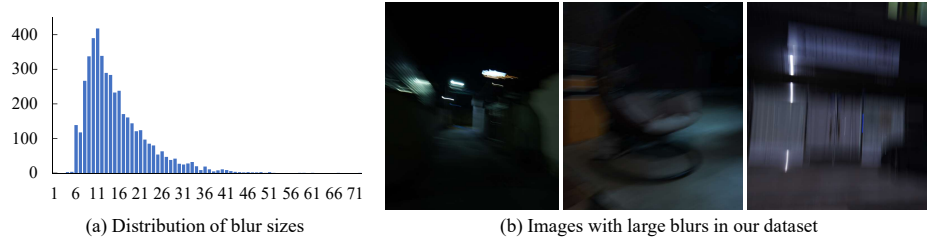
$$G(u, v) = F(u, v)H(u, v) \quad (6)$$

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:

$$S(u, v) = \frac{G(u, v)F^*(u, v)}{|G(u, v)F^*(u, v)|} = \frac{H(u, v)F(u, v)F^*(u, v)}{|H(u, v)F(u, v)F^*(u, v)|} = \frac{H(u, v)}{|H(u, v)|} \quad (7)$$

where  $H(u, v)/|H(u, v)|$  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. 1. 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 [7] or to spatially-varying blur. Nonetheless, thanks to our



**Fig. 2.** 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).

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.

## 2 Additional Analysis on Our 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. 2(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. 2(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 [1]. The table also reports the noise levels of the GoPro [6] and Reds [5] 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 [6] and Reds [5] 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.

## 3 Training Details

In this section, we describe how we train deblurring networks with our dataset in our experiment.

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

Dataset		Noise level
<i>RealBlur-R</i>	Sharp images before denoising	2.6714
	Sharp images after denoising	0.3016
	Blurred images	0.6654
<i>RealBlur-J</i>	Sharp images	0.3029
	Blurred images	0.4369
GoPro [6]	Sharp images	0.3474
	Blurred images	0.3042
Reds [5]	Sharp images	0.3571
	Blurred images	0.3868

**SRN-DeblurNet [10].** We use the source code of the authors provided on their project webpage<sup>3</sup>. 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 [10]. For training the network using all three datasets, which are ours, BSD-B and GoPro [6], we oversample our dataset six times, and the GoPro dataset 10 times to balance the datasets of different sizes.

**DeblurGAN-v2 [3].** We use the framework with Inception-ResNet-v2 as its backbone, and the double-scale discriminator with the RaGAN-LS loss [3]. We use the source code provided by the authors<sup>4</sup>. We train the network using the Adam optimizer with hyperparameters described in [3] except the number of iterations. Specifically, we train the network with the learning rate of  $10^{-4}$  for 1,500,000 iterations, then linearly decay the learning rate to  $10^{-7}$  for another 1,500,000 iterations. For training the network with our dataset and the BSD-B and GoPro [6] datasets, we oversample our dataset six times, and the GoPro dataset 10 times to balance the datasets of different sizes.

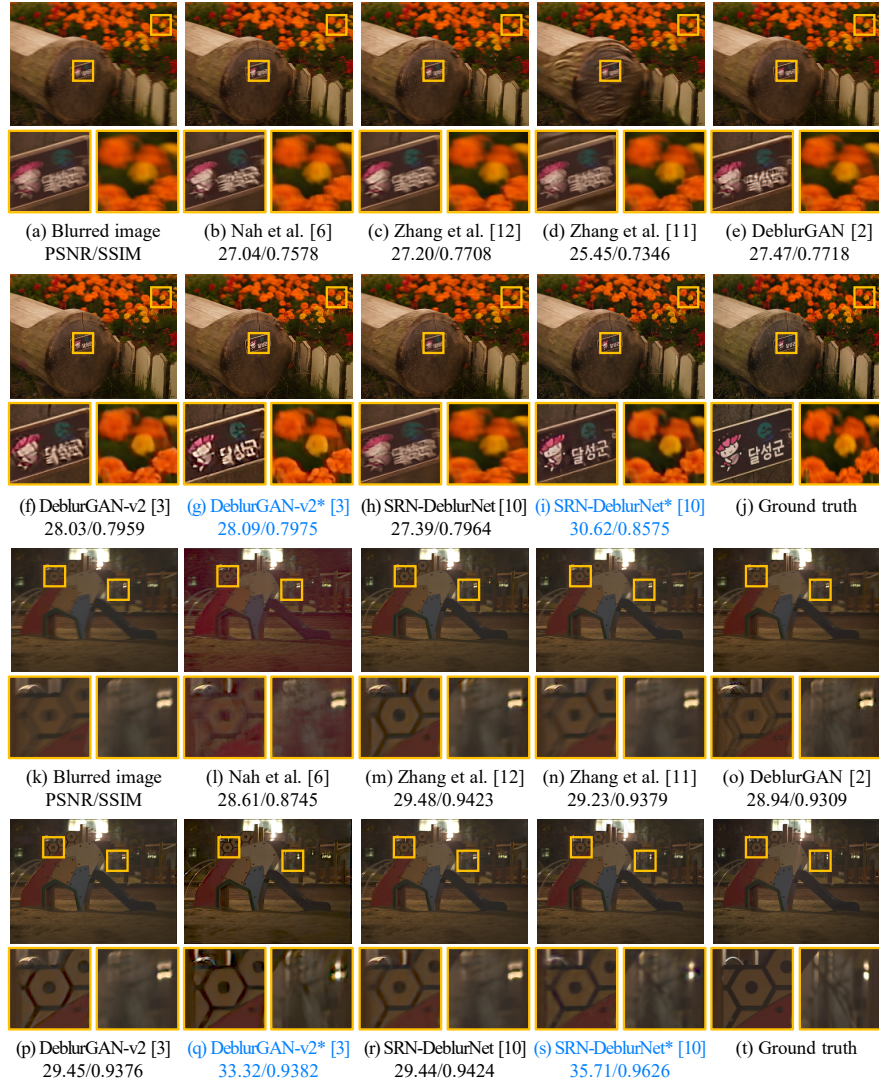
## 4 Generalization to Different Lenses

We present an additional experimental result that shows the generalization ability of deblurring networks trained with our dataset. Specifically, in this section, we present an experimental result on the performance of a deblurring network trained with our dataset on telephoto images. We refer the readers to the Supplementary Material II for other experimental results that show the generalization ability to other datasets captured with different cameras such as smartphone cameras.

<sup>3</sup> <https://github.com/jiangsutx/SRN-Deblur>

<sup>4</sup> <https://github.com/TAMU-VITA/DeblurGANv2>





**Fig. 3.** 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 [3] and SRN-DeblurNet [10] 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.

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

**Table 2.** 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.

<i>RealBlur-TeleJ</i>		<i>RealBlur-TeleR</i>	
Methods	PSNR/SSIM	Methods	PSNR/SSIM
SRN-DeblurNet* [10]	28.14/0.8463	SRN-DeblurNet* [10]	37.37/0.9470
DeblurGAN-v2* [3]	26.87/0.8041	DeblurGAN-v2* [3]	35.44/0.9112
DeblurGAN-v2 [3]	26.27/0.8020	SRN-DeblurNet [10]	35.36/0.9303
SRN-DeblurNet [10]	25.89/0.7943	Zhang <i>et al.</i> [12]	35.19/0.9263
Zhang <i>et al.</i> [12]	25.61/0.7794	DeblurGAN-v2 [3]	35.12/0.9258
DeblurGAN [2]	25.55/0.7676	Zhang <i>et al.</i> [11]	34.67/0.9216
Nah <i>et al.</i> [6]	25.37/0.7580	DeblurGAN [2]	33.01/0.8815
Zhang <i>et al.</i> [11]	24.72/0.7581	Nah <i>et al.</i> [6]	32.39/0.8419

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 of 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 [3] and SRN-DeblurNet [10] with *RealBlur-J* as well as the GoPro [6] 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. 3 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 2 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.

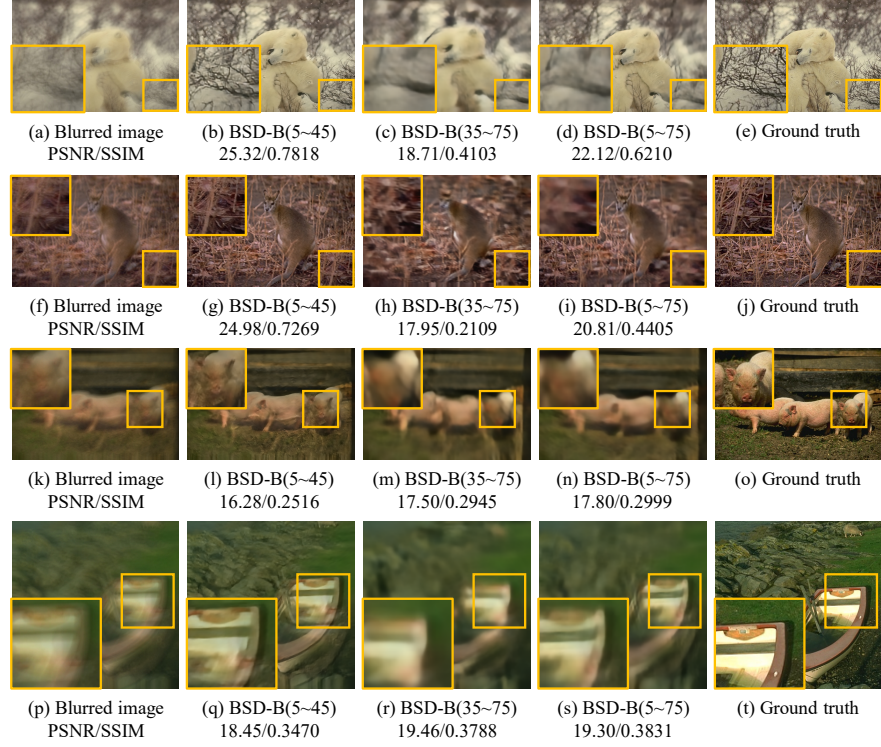
## 5 Deblurring Images with Large Blur

As shown in Sec. 2, our dataset mostly contains relatively small-sized blurs. In our preliminary experiments, we found that existing networks such as [6, 10] mostly fail to handle large blurs even if we train them with large blurs. Thus, for *RealBlur*, we mainly collected images with small blurs to study the domain gap between synthetic and real-world blurs. In this section, we describe one preliminary experiment to study the capability of existing methods on large blurs.

Most existing deblurring networks [6, 10, 12, 11] share common components such as the encoder-decoder and multi-scale structures. In our experiment, we evaluate only the performance of SRN-DeblurNet [10] on large blurs as it is a representative method that is equipped with the encoder-decoder and multi-scale structures.

For the experiment, we randomly selected 400 images for generating training sets and the remaining 100 images for generating test sets from the BSD500 dataset [4]. Then, we generated three training sets with different blur sizes using synthetic uniform blur kernels. Specifically, for each sharp image, we generated 40 synthetic blur kernels using Schmidt *et al.*'s method [9] to generate blurred images. We generated a small blur dataset with blur sizes from 5 to 45 pixels, and a large blur dataset with blur sizes from 35 to 75 pixels. In both datasets, blur sizes are uniformly distributed. Then, we merged the two datasets and obtain the third dataset that has both small and large blurs. We denote these training sets as  $\text{BSD-B}(5-45)_{\text{train}}$ ,  $\text{BSD-B}(35-75)_{\text{train}}$ , and  $\text{BSD-B}(5-75)_{\text{train}}$ , respectively. Each of  $\text{BSD-B}(5-45)_{\text{train}}$  and  $\text{BSD-B}(35-75)_{\text{train}}$  has 16,000 blurred images, while  $\text{BSD-B}(5-75)_{\text{train}}$  has 32,000 images. We generated two test sets with different blur sizes in a similar manner, which are denoted as  $\text{BSD-B}(5-45)_{\text{test}}$  and  $\text{BSD-B}(35-75)_{\text{test}}$ . Each of  $\text{BSD-B}(5-45)_{\text{test}}$  and  $\text{BSD-B}(35-75)_{\text{test}}$  has 4,000 blurred images. We trained SRN-DeblurNet on each training set and compared its performance on each test set.

Fig. 4 shows some examples of deblurring results of the models trained with different training sets. Table 3 reports the performance with respect to different training and test sets. As shown in Fig. 4, regardless of the training sets, the network often fails to deblur images with large blurs in  $\text{BSD-B}(35-75)_{\text{test}}$ , and achieves low PSNR and SSIM values, which indicates the limited capability of the network. Another interesting observation the figure and table is that the performance of the network trained with  $\text{BSD-B}(5-75)_{\text{train}}$  is lower than the network trained with  $\text{BSD-B}(5-45)_{\text{train}}$  on  $\text{BSD-B}(5-45)_{\text{test}}$ . This indicates that including images with large blurs in training hinders the performance of the network, which is another cue showing the limited capability of the network.



**Fig. 4.** Qualitative comparison of deblurring results with different blur sizes. The first column ((a), (f), (k), (p)) shows input blurred images with different blur sizes. The second, third, and fourth columns show deblurring results of SRN-DeblurNet [10] trained with  $\text{BSD-B}(5\sim45)_{\text{train}}$ ,  $\text{BSD-B}(35\sim75)_{\text{train}}$ , and  $\text{BSD-B}(5\sim75)_{\text{train}}$ , respectively. The fifth column shows the ground truth sharp images. The blur sizes of (a), (f), (k), and (p) are 21, 15, 71, and 69, respectively.

**Table 3.** Quantitative comparison of SRN-DeblurNet [10] trained with training sets with different blur sizes.

Training sets	Test sets (PSNR/SSIM)	
	$\text{BSD-B}(5\sim45)_{\text{test}}$	$\text{BSD-B}(35\sim75)_{\text{test}}$
$\text{BSD-B}(5\sim45)_{\text{train}}$	25.65/0.7098	19.71/0.4640
$\text{BSD-B}(35\sim75)_{\text{train}}$	22.35/0.5860	21.21/0.5152
$\text{BSD-B}(5\sim75)_{\text{train}}$	25.17/0.6918	21.55/0.5243

## References

1. Chen, G., Zhu, F., Ann Heng, P.: An efficient statistical method for image noise level estimation. In: ICCV (December 2015)
2. Kupyn, O., Budzan, V., Mykhailych, M., Mishkin, D., Matas, J.: DeblurGAN: Blind motion deblurring using conditional adversarial networks. In: CVPR (June 2018)

3. Kupyn, O., Martyniuk, T., Wu, J., Wang, Z.: DeblurGAN-v2: Deblurring (orders-of-magnitude) faster and better. In: ICCV (October 2019)
4. 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)
5. 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)
6. Nah, S., Hyun Kim, T., Mu Lee, K.: Deep multi-scale convolutional neural network for dynamic scene deblurring. In: CVPR (July 2017)
7. 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)
8. 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)
9. 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>
10. Tao, X., Gao, H., Shen, X., Wang, J., Jia, J.: Scale-recurrent network for deep image deblurring. In: CVPR (June 2018)
11. Zhang, H., Dai, Y., Li, H., Koniusz, P.: Deep stacked hierarchical multi-patch network for image deblurring. In: CVPR (June 2019)
12. 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)