Supplemental Materials for "K3DN: Disparity-aware Kernel Estimation for Dual-Pixel Defocus Deblurring"

In this supplementary material, we provide additional implementation details (Appendix A), additional experiment results (Appendix B), additional ablation study (Appendix C), and limitations (Appendix D) for our K3DN framework.

A. Additional Implementation Details

K3DN uses a 3-level U-net architecture. We use AdamW optimizer [13] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, learning rate $= 3 \times 10^{-4}$, and weight decay $= 10^6$. We use the 'cosine annealing with warmups' learning rate scheduler, and set the 'cycle steps', 'warmup steps', and 'minimum learning rate' to 200, 100, and 6×10^{-5}. For the DPD-blur dataset [1], our model is trained for 20k iterations in a two-stage manner. First, we train our model without the SRP blocks from scratch for 9.8K iterations. Second, we freeze all model weights and train the newly added parameters from the SRP blocks for another 10.2K iterations, while excluding the reblurring loss $\mathcal{L}_{reb}$ from the overall training loss $\mathcal{L}$ as our target is to preserve the sharp regions of defocus blurred DP pair. For the DDD-syn dataset [15] and RDPD dataset [2], we adopt resource-constrained training, as the synthetic datasets are easy to be overfitted. Specifically, our model is respectively trained for 4k and 40k iterations on the two datasets. When the performance of other methods is not available, we train them with the same iterations for a fair comparison.

Our $\mathcal{L}_{deb}$ uses a combination of Multi-Scale Charbonnier loss $\mathcal{L}_{chb}$ [25], Multi-Scale Edge loss $\mathcal{L}_{edg}$ [25] and Multi-Scale Frequency loss $\mathcal{L}_{frq}$ [14], i. e., $\mathcal{L}_{deb} = \mathcal{L}_{chb} + \lambda_2\mathcal{L}_{edg} + \lambda_3\mathcal{L}_{frq}$. Meanwhile, we define $\mathcal{L}_{reb}$ as a mean squared error-based loss. We set $\lambda_1 = 1 \times 10^{-1}$, and $\lambda_2 = 5 \times 10^{-2}$, $\lambda_3 = 1 \times 10^{-2}$. During optimization, we apply gradient norm clipping at 1×10^{-2}.

The detailed architecture of our K3DN framework is summarised in Tab. 13. All convolution layers apply a LeakyReLU with a negative slope of 0.2. We use `Num` as a column attribute to represent the number of replication for current layers. We denote `bn` and `bc` as the base number of replication and base channel. The configurations of our model variants (i. e., Tiny, Lightweight, and Large) are in Tab. 6.

Table 6. *Configurations of different model variants.*

Variants	bn	bc
Tiny	2	24
Lightweight	2	32
Large	4	48

Figure 9. *Samples of reblurred images (zoom in for better quality).*

B. Additional Experiment Results

We briefly investigate the reblur capability of our model in Fig. 9. Next, we verify our model generalization ability in Fig. 11. We then study our disparity estimator that is trained in an unsupervised manner, in Fig. 12. In the following, we visualize the sub-kernel with the largest weight assigned by the disparity vector for different image regions in our PSF block (Fig. 13). Note that we linearly transform the image space to feature space and train a PSF block for better kernel visualization. Finally, we present more comparisons with state-of-the-art methods (Fig. 14, Fig. 15, Fig. 16, Fig. 17 and Fig. 18), in addition to the Fig. 6 and Fig. 7 from our main paper. Specifically, we compare with RDPD [2], KPAC [19], IFAN [10], DeepRFT [14], DDDNet [15], RDPD [2], BAMBNet [11], and Restormer [24]. Note that we use their publicly available checkpoint to generate the all-in-focus restorations.

We also test our model on Google Pixels dataset [23] in

Table 7. *Performance evaluation on Google Pixels DP image dataset from [23]. The performance of our tiny model is presented.*

Model	PSNR↑	SSIM↑	RMSE_rel$_{(10^{-2})}$ ↓	MAE$_{(10^{-1})}$ ↓
Wiener Deconv [27]	25.81	0.704	5.13	0.320
DPDNet [1]	25.59	0.777	5.25	0.340
Xin *et al.* [23]	26.69	0.804	4.93	0.270
IFAN [10]	31.49	0.867	2.66	0.164
Restormer [24]	31.27	0.859	2.73	0.161
Ours	31.59	0.891	2.63	0.165

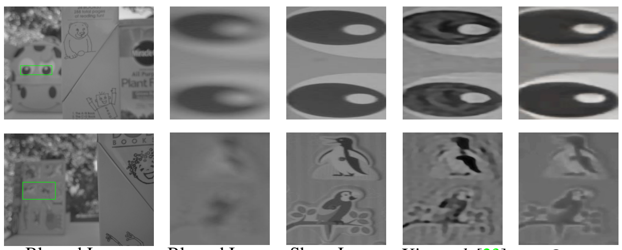


Figure 10. *Comparison of image restoration performance on the Google Pixels dataset [23].*

Table 8. *Single image defocus deblurring of our method on the DPD-blur dataset.*

Model	PSNR↑	SSIM↑	RMSE_rel$_{(10^{-2})}$ ↓	MAE$_{(10^{-1})}$ ↓
Ours (Tiny)	25.85	0.794	5.10	0.380
Ours (Lightweight)	25.95	0.799	5.04	0.377
Ours (Large)	26.11	0.805	4.95	0.372

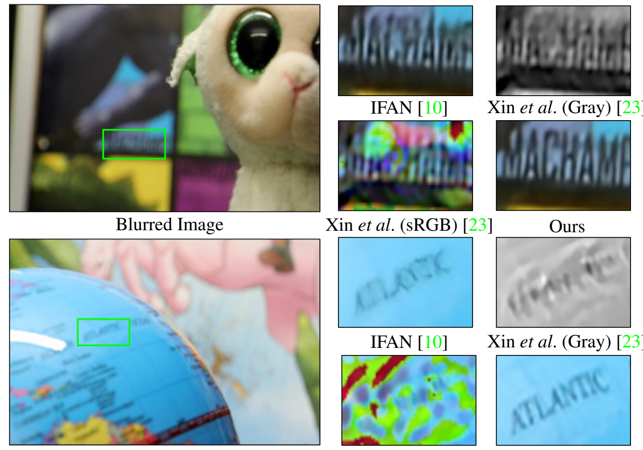


Figure 11. *The generalization ability of disparity-based methods (an expansion of Fig. 7). Here, we mainly consider the disparity-based approaches, i. e., IFAN [10] and Xin et al. [23] (refer to Fig. 7 for restoration results of other methods). Note that all methods are not trained and specialized for the DPD-disp dataset [16], i. e., our model and IFAN use the pretrained checkpoint on the DPD-blur dataset [1], and Xin et al. uses the provided and pre-calibrated kernels. We present two kinds (Gray and sRGB) of restored images for Xin et al. [23], where the sRGB restored images are generated by deblurring on each channel independently.*

Tab. 7 and Fig. 10. Note that the brightness and contrast of restorations are adjusted for better visualization. This dataset is captured by Google Pixels smartphone, and provides 17 pairs of defocus blurred DP images and associated all-in-focus images. It covers both indoor and outdoor scenes. We test K3DN framework by using the pretrained checkpoint on the DPD-blur dataset. Similarly, we present the performance of Restormer [24] and IFAN [10], the latest state-of-the-art method, in this dataset.

Moreover, we adapt our K3DN framework to perform the single image defocus deblurring task (i. e., use the center view of the DP image) on the DPD-blur dataset. The performance is presented in Tab. 8.

C. Additional Ablation Study

All ablation studies are conducted with our lightweight model.

The alignment of encoder and disparity estimator. As discussed in Sec. 3, $\mathbf{F}_{\mathrm{B}}$ and $\mathbf{R}$ are spatially aligned with each other, while each i-th layer features of $\mathbf{F}_{\mathrm{B}}$ can be founded by performing a nearest neighbor interpolation. In other words, each vector $\mathbf{r}^i \in \mathbf{R}$ is spatially aligned with $\mathbf{F}^i_B \in \mathbf{F}_B$. By varying the downsampling rate (*e.g.*, the stride of convolution) and resizing the inputs for our disparity estimator, for each $\mathbf{r}^i$, the spatial size (i. e., $\frac{\mathsf{H_f}}{\mathsf{H_d}} \times \frac{\mathsf{W_f}}{\mathsf{W_d}}$) of the aligned feature $\mathbf{F}^i_B$ is changed accordingly. Here, we study the impact (Tab. 9) of $\frac{\mathsf{H_f}}{\mathsf{H_d}} \times \frac{\mathsf{W_f}}{\mathsf{W_d}}$ in the DPD-blur dataset [1].

Table 9. *Alignment of encoder and disparity estimator.*

$\frac{\mathsf{H_f}}{\mathsf{H_d}} \times \frac{\mathsf{W_f}}{\mathsf{W_d}}$	9×9	14×18	18×14	18×18	27×27
PSNR↑	26.76	26.77	26.84	26.72	26.60

With $\frac{\mathsf{H_f}}{\mathsf{H_d}} = 18$ and $\frac{\mathsf{W_f}}{\mathsf{W_d}} = 14$, we find the best performance. This is potentially determined by the complexity of the blur model in the dataset. During testing, to be compatible with diverse sizes of model inputs, we resize the inputs to the multiples of the spatial size, and then we rescale the model outputs to the original size.

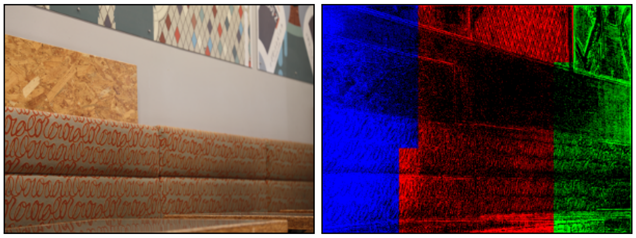
Spatial size of the kernel set. We analyze the spatial size of the kernel set (i. e., $\mathsf{H_k} \times \mathsf{W_k}$) of the candidate kernel set $\mathcal{K}$ in Tab. 10.

Table 10. *Impact of the spatial size of the kernel set.*

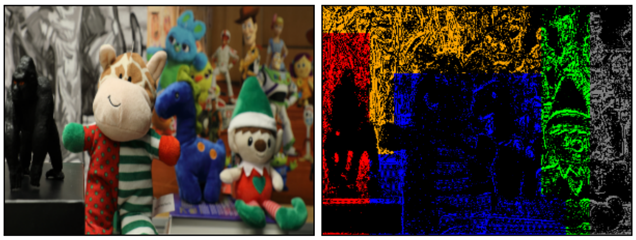
$\mathrm{H_k} \times \mathrm{W_k}$	3×3	5×5	7×7	9×9	11×11	13×13
PSNR↑	26.81	26.79	26.77	26.84	26.79	26.76

Considering the model performance, we set $\mathrm{H_k} = 9$ and $\mathrm{W_k} = 9$.

Number of the PSF blocks. By fixing all other components of a lightweight K3DN framework, we study the optimal number of PSF blocks in Tab. 11. Note that the



(a)



(b)

Figure 12. *(a)-(b) Examples of input left view DP images and their associated disparity feature clusters. With obtained features from our disparity estimator, we perform a k-means algorithm to cluster similar disparity features across the image. The assigned cluster-IDs are used to colorize the latent features processed by the PSF block.*

Figure 13. *Sample kernels from the PSF block.*

lightweight K3DN has 4 PSF blocks (i.e., 2 × `bn` in Appendix A and Tab. 13).

Table 11. *Number of PSF blocks.*

#PSF blocks	1	2	3	4	5	6	7
PSNR↑	26.69	26.72	26.76	26.84	26.81	26.77	26.73

Conceptually, the more PSF blocks, the more complex blur models that we can handle. However, with the lightweight model size, there is limited feature semantics that can be embedded in the feature space due to the small model size. Therefore, a large number of PSF blocks can potentially harm the model generalization ability, and we find the optimal number of PSF blocks is 4.

Inference speed. We investigate the inference speed of K3DN and other state-of-the-art methods in Tab. 12. The experiments are conducted under a single NVIDIA A40 GPU. We use batch size 1, warm up the GPU for 5 iterations, and average 30 random testing results. In comparison to the latest state-of-the-art method, Restormer [24], our method has significant inference speed improvements without any performance deterioration (refer to Tab. 1, Tab. 2 and Tab. 3 for the performance comparison).

Table 12. *Inference speed of past methods.*

Method	Restormer	BAMBNet	DeepRFT	DRBNet
Second	2.38	0.970	1.03	0.197
Method	IFAN	Ours (Tiny)	Ours (Lightweight)	Ours (Large)
Second	0.142	0.236	0.318	0.578

D. Limitations

Though our PSF blocks follow the blur mode of the DP image formulation (Sec. 3.1) and our K3DN framework achieves a favorable deblur performance, the exact inversion for the model is not maintained. For example, in the deblurring and reblurring processes, our encoder and decoder do not have an exact inverse constraint (i. e., they are trained to perform encoding and decoding), and only the inversion within each PSF block is maintained. In our future work, we plan to study fully invertible network architectures for K3DN.

Table 13. *K3DN architecture. We use ↓ and ↑ to denote downsampling and upsampling, respectively. For the PSF block, a point-wise convolution [20] and a residual connection are also added to improve the feature representation ability, where the kernel sizes are specified accordingly. Note that a point-wise convolution is easy to invert by using the LU decomposition [8].*

	Type	Input	Activation	Kernel	Channel	Stride	Padding	Dilation	Num	Output
Disparity Estimator	FE	$\mathbf{B}_{\mathrm{L}\downarrow 4}$	-	-	-	-	-	-	1	lt
	FE	$\mathbf{B}_{\mathrm{R}\downarrow 4}$	-	-	-	-	-	-	1	rt
	cost	{lt, rt}	-	-	-	-	-	-	1	c_1
	conv3d	c_1	ReLU	3	32	1	1	1	1	c_2
	conv3d	c_2	ReLU	3	48	2	1	1	1	c_3
	conv3d	c_2	ReLU	3	48	1	1	1	1	c_4
	conv3d	c_4	ReLU	3	64	2	1	1	1	c_5
	conv3d	c_5	ReLU	3	64	1	1	1	1	c_6
	Reshape, Pooling based on Patch Size, and Linear projection.									$\mathbf{R}$
↓ Shared Feature Extractor.										
FE (Feature Extractor)	*conv*	Input	ReLU	3	32	1	1	1	1	b_1
	conv	b_1	ReLU	3	64	1	1	1	1	b_2
	conv	b_2	ReLU	3	128	1	4	4	1	b_3
	conv	b_3	ReLU	3	128	1	8	8	1	b_4
	AvgPool	b_4	-	16	-	16	-	-	-	b_5
	conv	b_5	ReLU	3	32	1	1	1	1	b_6
	AvgPool	b_4	-	32	-	32	-	-	-	b_7
	conv	b_7	ReLU	3	32	1	1	1	1	b_9
	conv	$\{\mathrm{b}_4, \{\mathrm{b}_6\}_{\uparrow 16}, \{\mathrm{b}_8\}_{\uparrow 32}\}$	ReLU	3	96	1	1	1	1	b_9
	conv	b_9	ReLU	3	32	1	1	1	1	b_{10}
↓ Deblurring Framework. Shared Encoder, PSF Blocks, and Decoder.										
Encoder	*conv*	$\{\mathbf{B}_\mathrm{L}, \mathbf{B}_\mathrm{R}\}$	LeakyReLU	3	bc	2	1	1	1	d_1
	res	d_1	LeakyReLU	3	bc	1	1	1	bn	d_2
	conv	d_2	LeakyReLU	3	2×bc	2	1	1	1	d_3
	res	d_3	LeakyReLU	3	2×bc	1	1	1	bn	d_4
	conv	d_4	LeakyReLU	3	4×bc	2	1	1	1	d_5
	res	d_5	LeakyReLU	3	4×bc	1	1	1	bn	$\mathbf{F}_\mathbf{B}$
	PSF	$\{\mathbf{F}_\mathbf{B}, \mathbf{R}\}$	-	{9, 1}	4×bc	1	{5, 1}	1	2×bn	$\hat{\mathbf{F}}_\mathbf{B}$
Decoder	*dconv*	$\hat{\mathbf{F}}_\mathbf{B}$	LeakyReLU	4	2×bc	2	1	1	1	u_1
	SRP	$\{\mathrm{d}_4, \mathrm{u}_1\}$	LeakyReLU	3	2×bc	1	1	1	1	s_1
	res	s_1	LeakyReLU	3	4×bc	1	1	1	bn	u_2
	dconv	u_2	LeakyReLU	4	2×bc	2	1	1	1	u_3
	SRP	$\{\mathrm{d}_2, \mathrm{u}_3\}$	LeaklyReLU	3	2×bc	1	1	1	1	s_2
	res	s_2	LeakyReLU	3	2×bc	1	1	1	bn	u_4
	dconv	u_4	LeakyReLU	4	bc	2	1	1	1	u_5
	SRP	$\{\mathbf{B}_\mathrm{L}, \mathbf{B}_\mathrm{R}, \mathrm{u}_5\}$	LeaklyReLU	3	bc+6	1	1	1	1	s_3
	res	s_3	LeakyReLU	3	bc+6	1	1	1	bn	u_6
	conv	u_6	-	3	3	1	1	1	1	$\mathbf{I}$
↓ Reblurring Framework. Shared Encoder, PSF Blocks, and Decoder.										
Encoder	*conv*	$\{\mathbf{I}, \mathbf{I}\}$	LeakyReLU	3	bc	2	1	1	1	d_1
	res	d_1	LeakyReLU	3	bc	1	1	1	bn	d_2
	conv	d_2	LeakyReLU	3	2×bc	2	1	1	1	d_3
	res	d_3	LeakyReLU	3	2×bc	1	1	1	bn	d_4
	conv	d_4	LeakyReLU	3	4×bc	2	1	1	1	d_5
	res	d_5	LeakyReLU	3	4×bc	1	1	1	bn	$\mathbf{F}_\mathbf{I}$
	PSF	$\{\mathbf{F}_\mathbf{I}, \mathbf{R}\}$	-	{9, 1}	4×bc	1	{5, 1}	1	2×bn	$\hat{\mathbf{F}}_\mathbf{I}$
Decoder	*dconv*	$\hat{\mathbf{F}}_\mathbf{I}$	LeakyReLU	4	2×bc	2	1	1	1	u_1
	res	$\{\mathrm{d}_4, \mathrm{u}_1\}$	LeakyReLU	3	4×bc	1	1	1	bn	u_2
	dconv	u_2	LeakyReLU	4	2×bc	2	1	1	1	u_3
	res	$\{\mathrm{d}_2, \mathrm{u}_3\}$	LeakyReLU	3	2×bc	1	1	1	bn	u_4
	dconv	u_4	LeakyReLU	4	bc	2	1	1	1	u_5
	res	$\{\mathbf{I}, \mathbf{I}, \mathrm{u}_5\}$	LeakyReLU	3	bc+6	1	1	1	bn	u_6
	conv	u_6	-	3	3	1	1	1	1	$\mathbf{B}$

Figure 14. *Comparison of image restoration performance on the DPD-blur dataset [1]. The large sharp images in the first column are ground-truth sharp images. The small sharp images in the second column are cropped images from the green bounding box in the large ground-truth sharp images. The blurred images in the second column are corresponding input blurry images ($\mathbf{B}_\mathrm{L}$).*

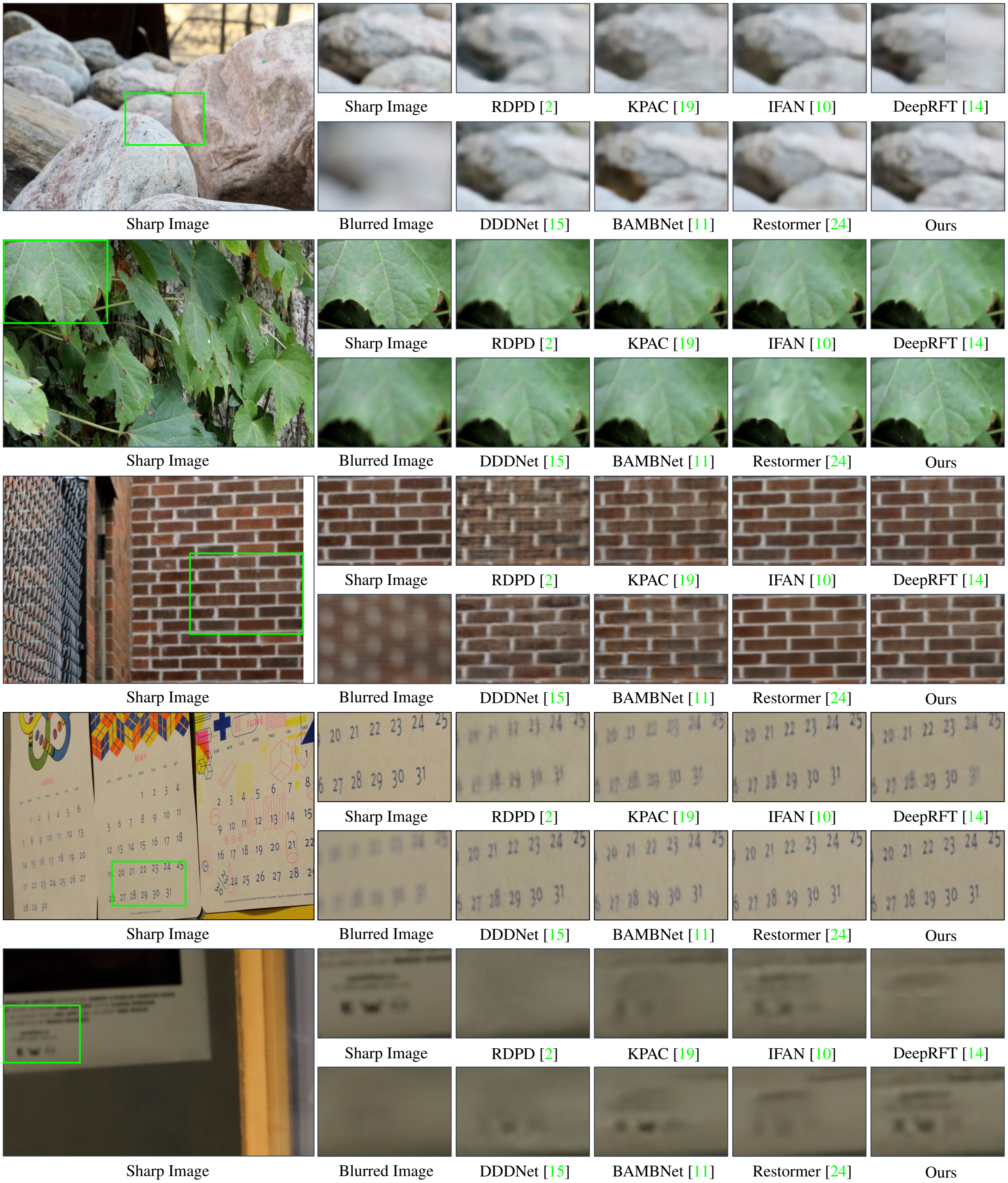


Figure 15. *Comparison of image restoration performance on the DPD-blur dataset [1]. The large sharp images in the first column are ground-truth sharp images. The small sharp images in the second column are cropped images from the green bounding box in the large ground-truth sharp images. The blurred images in the second column are corresponding input blurry images ($\mathbf{B}_{\mathrm{L}}$).*

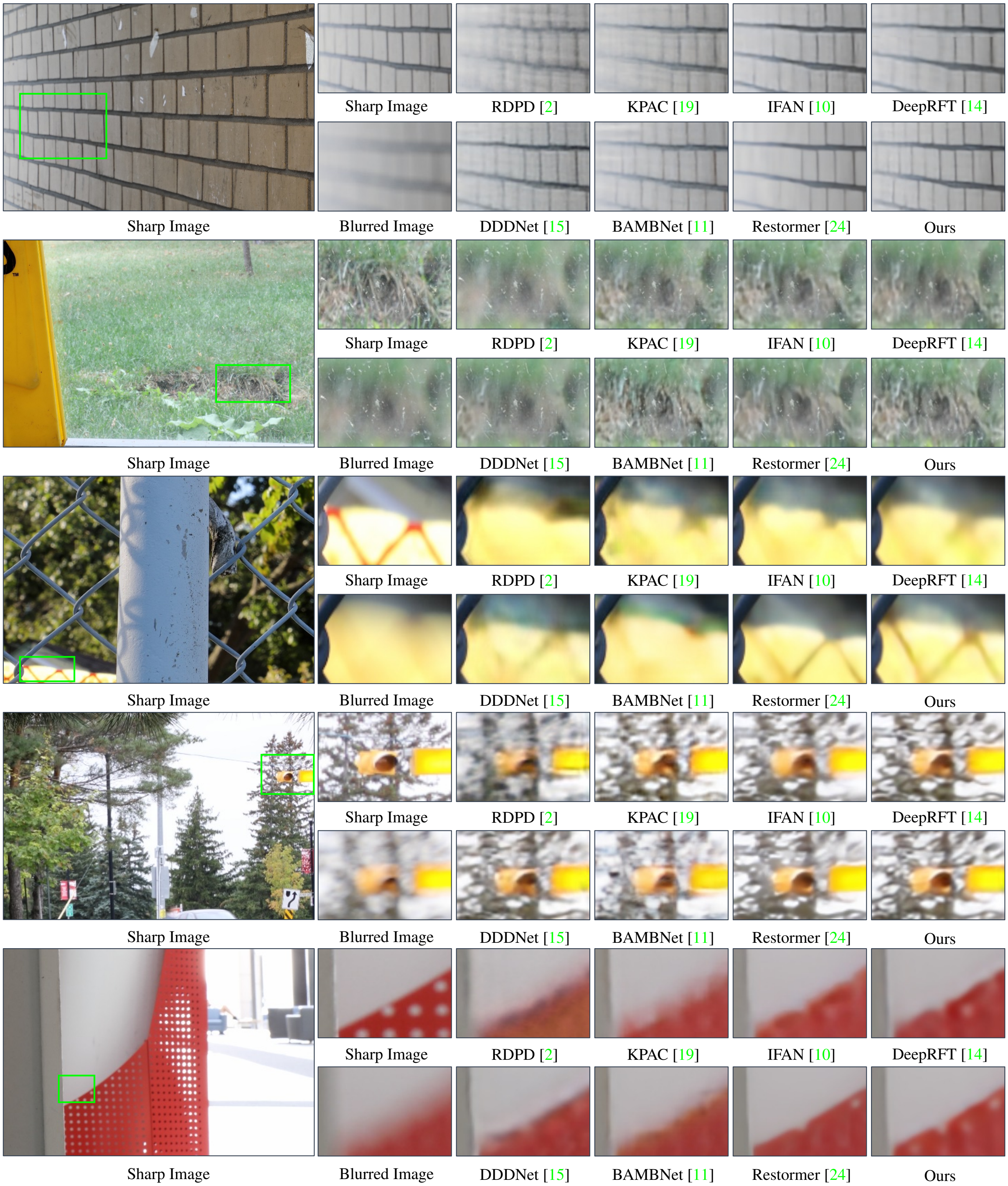


Figure 16. *Comparison of image restoration performance on the DPD-blur dataset [1]. The large sharp images in the first column are ground-truth sharp images. The small sharp images in the second column are cropped images from the green bounding box in the large ground-truth sharp images. The blurred images in the second column are corresponding input blurry images ($\mathbf{B}_{\mathrm{L}}$).*

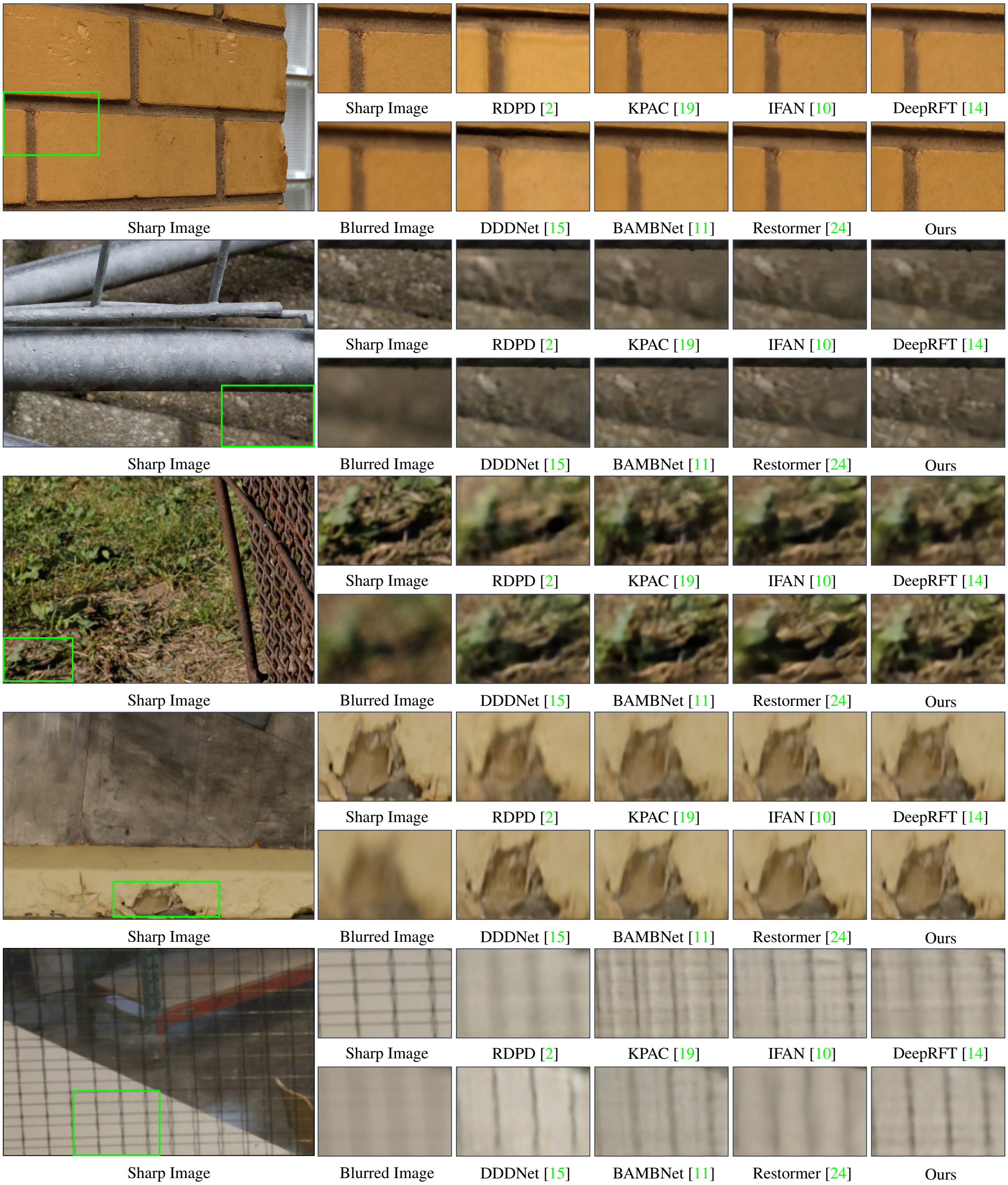


Figure 17. *Comparison of image restoration performance on the DPD-blur dataset [1]. The large sharp images in the first column are ground-truth sharp images. The small sharp images in the second column are cropped images from the green bounding box in the large ground-truth sharp images. The blurred images in the second column are corresponding input blurry images ($\mathbf{B}_\mathrm{L}$).*

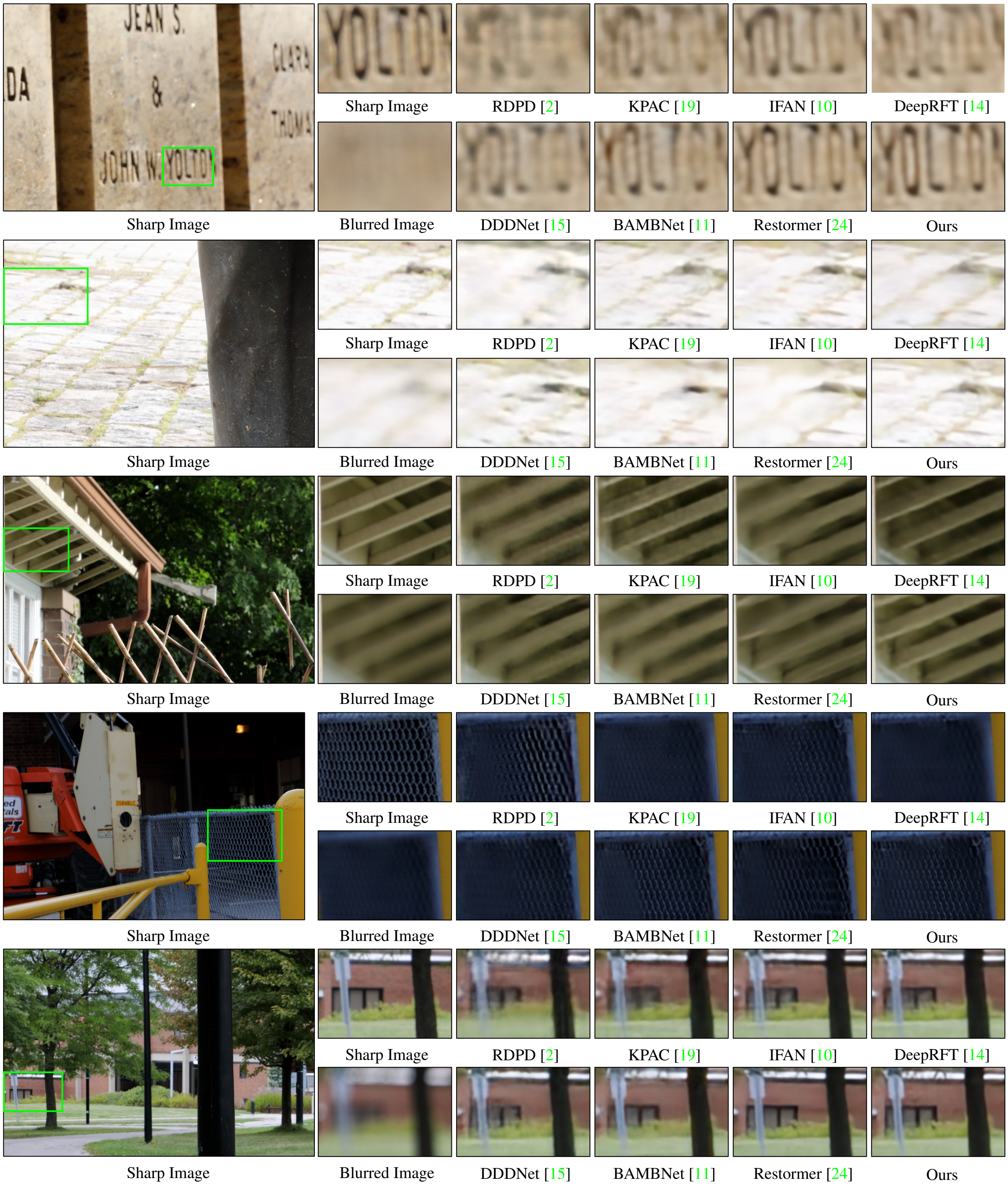


Figure 18. *Comparison of image restoration performance on the DPD-blur dataset [1]. The large sharp images in the first column are ground-truth sharp images. The small sharp images in the second column are cropped images from the green bounding box in the large ground-truth sharp images. The blurred images in the second column are corresponding input blurry images ($\mathbf{B}_{\mathrm{L}}$).*

References

[1] Abdullah Abuolaim and Michael S Brown. Defocus deblurring using dual-pixel data. In *European Conference on Computer Vision*, pages 111–126. Springer, 2020. 1, 2, 5, 6, 7, 8, 11, 12, 15, 16, 17, 18, 19

[2] Abdullah Abuolaim, Mauricio Delbracio, Damien Kelly, Michael S Brown, and Peyman Milanfar. Learning to reduce defocus blur by realistically modeling dual-pixel data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2289–2298, 2021. 1, 2, 5, 6, 7, 8, 11, 15, 16, 17, 18, 19

[3] Xuelian Cheng, Yiran Zhong, Mehrtash Harandi, Yuchao Dai, Xiaojun Chang, Hongdong Li, Tom Drummond, and Zongyuan Ge. Hierarchical neural architecture search for deep stereo matching. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 4

[4] Rahul Garg, Neal Wadhwa, Sameer Ansari, and Jonathan T. Barron. Learning single camera depth estimation using dual-pixels. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 7627–7636. IEEE, 2019. 3

[5] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc Van Gool. Dynamic filter networks. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 667–675, 2016. 2

[6] Daniel Hernández Juárez, Lukas Schneider, Antonio Espinosa, Juan C. Moure, David Vázquez, Antonio M. López, Uwe Franke, and Marc Pollefeys. Slanted stixels: Representing san francisco's steepest streets. In *British Machine Vision Conference 2017, BMVC 2017, London, UK, September 4-7, 2017*. BMVA Press, 2017. 5

[7] Ali Karaali and Claudio Rosito Jung. Edge-based defocus blur estimation with adaptive scale selection. *IEEE Transactions on Image Processing*, 27(3):1126–1137, 2017. 6

[8] Diederik P. Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 10236–10245, 2018. 14

[9] Junyong Lee, Sungkil Lee, Sunghyun Cho, and Seungyong Lee. Deep defocus map estimation using domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12222–12230, 2019. 6

[10] Junyong Lee, Hyeongseok Son, Jaesung Rim, Sunghyun Cho, and Seungyong Lee. Iterative filter adaptive network for single image defocus deblurring. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 2034–2042. Computer Vision Foundation / IEEE, 2021. 1, 2, 5, 6, 7, 8, 11, 12, 15, 16, 17, 18, 19

[11] Pengwei Liang, Junjun Jiang, Xianming Liu, and Jiayi Ma. Bambnet: A blur-aware multi-branch network for defocus deblurring. *CoRR*, abs/2105.14766, 2021. 1, 2, 6, 7, 8, 11, 15, 16, 17, 18, 19

[12] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. 6

[13] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. 11

[14] Xintian Mao, Yiming Liu, Wei Shen, Qingli Li, and Yan Wang. Deep residual fourier transformation for single image deblurring. *CoRR*, abs/2111.11745, 2021. 1, 6, 7, 8, 11, 15, 16, 17, 18, 19

[15] Liyuan Pan, Shah Chowdhury, Richard Hartley, Miaomiao Liu, Hongguang Zhang, and Hongdong Li. Dual pixel exploration: Simultaneous depth estimation and image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4340–4349, June 2021. 1, 2, 5, 6, 7, 8, 11, 15, 16, 17, 18, 19

[16] Abhijith Punnappurath, Abdullah Abuolaim, Mahmoud Afifi, and Michael S. Brown. Modeling defocus-disparity in dual-pixel sensors. In *2020 IEEE International Conference on Computational Photography, ICCP 2020, Saint Louis, MO, USA, April 24-26, 2020*, pages 1–12. IEEE, 2020. 1, 2, 3, 5, 8, 12

[17] Lingyan Ruan, Bin Chen, Jizhou Li, and Miu-Ling Lam. Learning to deblur using light field generated and real defocus images. *CoRR*, abs/2204.00367, 2022. 1, 2, 6

[18] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from RGBD images. In *European Conference on Computer Vision*, pages 746–760. Springer, 2012. 5

[19] Hyeongseok Son, Junyong Lee, Sunghyun Cho, and Seungyong Lee. Single image defocus deblurring using kernel-sharing parallel atrous convolutions. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 2622–2630. IEEE, 2021. 1, 6, 7, 8, 11, 15, 16, 17, 18, 19

[20] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 6105–6114. PMLR, 2019. 14

[21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. 06 2017. 2, 4

[22] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibil-

ity to structural similarity. *IEEE Trans. Image Process.*, 13(4):600–612, 2004. 5

[23] Shumian Xin, Neal Wadhwa, Tianfan Xue, Jonathan T. Barron, Pratul P. Srinivasan, Jiawen Chen, Ioannis Gkioulekas, and Rahul Garg. Defocus map estimation and deblurring from a single dual-pixel image. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 2208–2218. IEEE, 2021. 1, 2, 3, 5, 11, 12

[24] Syed Waqas Zamir, Aditya Arora, Salman H. Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. *CoRR*, abs/2111.09881, 2021. 1, 2, 6, 7, 8, 11, 12, 13, 15, 16, 17, 18, 19

[25] Syed Waqas Zamir, Aditya Arora, Salman H. Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 14821–14831. Computer Vision Foundation / IEEE, 2021. 11

[26] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VII*, volume 11211 of *Lecture Notes in Computer Science*, pages 294–310. Springer, 2018. 2

[27] Changyin Zhou, Stephen Lin, and Shree Nayar. Coded aperture pairs for depth from defocus. pages 325–332, 09 2009. 12