Learning to Reduce Defocus Blur by Realistically Modeling Dual-Pixel Data Supplemental Material

Abdullah Abuolaim[1*] Mauricio Delbracio[2] Damien Kelly[2] Michael S. Brown[1]
Peyman Milanfar[2]
[1]York University [2]Google Research

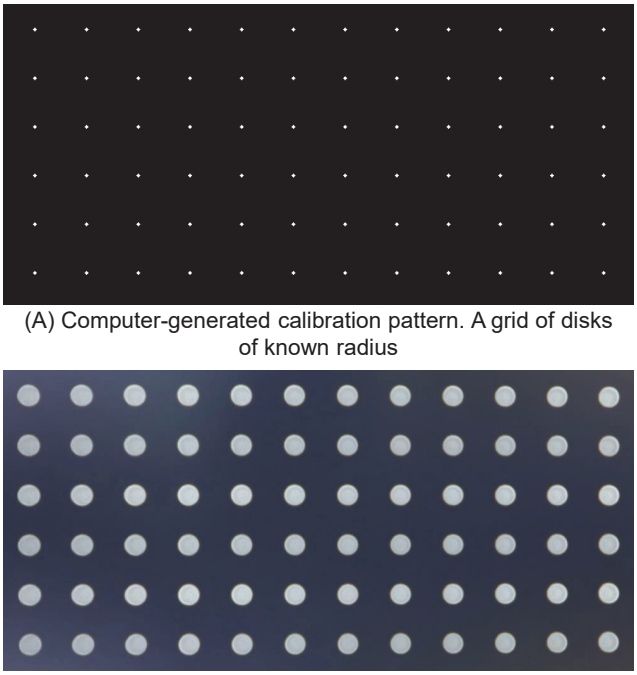
This supplemental material introduces the calibration procedure used to estimate estimate the point spread functions (PSFs) (Sec. S1) along with the parameter range searching for our parametric dual-pixel (DP) PSF model (Sec. S2). Next, another calibration procedure is described for estimating the radial distortion coefficients (Sec. S3). Afterward, an ablation study is provided to demonstrate the effectiveness of each component added to our proposed recurrent dual-pixel deblurring architecture (RDPD). Then, Sec. S5 provides a discussion about the difference between the defocus and motion blur. Additional qualitative results for different deblurring methods are also presented in Sec. S6.

For further visual assessment, we provide animated examples that alternate between the left and right DP views of real and synthetic data in the "animated-dp-views" directory. We also provide animated examples of our deblurring results in the "animated-results" directory. Both "animated-dp-views" and "animated-results" are located in the project GitHub repository: https://github.com/Abdullah-Abuolaim/recurrent-defocus-deblurring-synth-dual-pixel.

S1. PSFs on a real dual-pixel sensor

As mentioned in Sec. 2.2 of the main paper, we aim to estimate more accurate and realistic DP-PSFs; thus, we follow the same calibration practice as in [4, 8, 9]. The calibration pattern contains a grid of small disks with known radius and spacing as shown in Fig. S1-A. The pattern in Fig. S1-A is computer-generated, and we display it on a 27-inch LED display of resolution 1920×1080. Then, we use the Canon 5D Mark IV DSLR camera for our capturing procedure, as it facilitates reading out DP data. The LED display is placed parallel to the image plane and at a fixed distance of about one meter.

We captured many images by varying the camera parameters – namely, focus distance, aperture size, and focal length – covering a wide range of shape varying PSFs (an example image is shown in Fig. S1-B). Since we apply radial distortion to our synthetically generated data, we seek to estimate the least radially-distorted PSF, that in practice, is found close to the image center. Patches containing the disks are identified, and the center of the disks is estimated by finding these patches' centroid. The radius of computer-generated disks is a known fraction of the distance between disk centers.



(A) Computer-generated calibration pattern. A grid of disks of known radius

(B) Out-of-focus calibration pattern as imaged by Canon 5D Mark IV DSLR camera

Figure S1. Calibration patterns are used for estimating dual-pixel (DP) point spread functions (PSFs). A: our synthesized computer-generated pattern of a grid of small equal-size disks. B: the same calibration pattern as imaged by Canon 5D Mark IV DSLR camera. Note that the pattern captured in (B) is out-of-focus.

Similar to [8, 9], we adopt a non-blind PSF estimation, in which the latent sharp disk $\mathbf{S}$ is known. Then, the PSF from the camera $\mathbf{E}$ is estimated using $\mathbf{S}$ and the corresponding

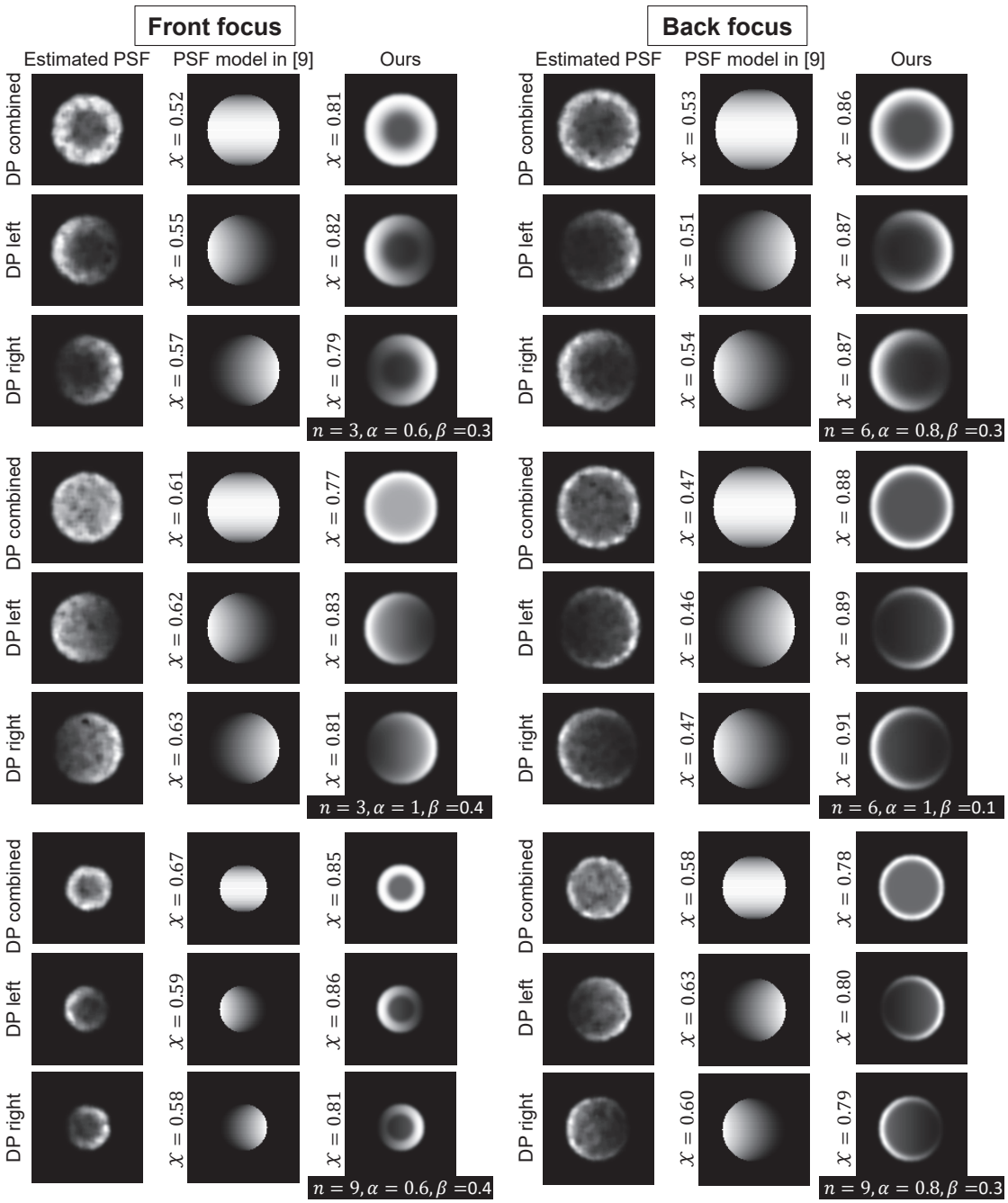


Figure S2. The estimated point spread functions (PSFs) in comparison with the dual-pixel (DP) PSF model in [9] and our parametric DP-PSF model. The PSFs for the DP combined, left, and right are estimated independently. The similarity with the estimated PSF is measured using the 2D cross-correlation $\mathcal{X}(.)$ and is shown on the left for each case. The parameters (i.e., n, α, β) used to generate our DP-PSF are shown below each case. Our parametric DP-PSF model obtains higher similarity with the estimated ones.

blurred patch $\mathbf{B}$ by solving:

$$\arg\min_{\mathbf{E}} \sum_i \left\| \mathbf{D}_i(\mathbf{S} * \mathbf{E} - \mathbf{B}) \right\|_2^2 + \left\| \mathbf{E} \right\|_1, \quad \text{subject to } \mathbf{E} \geq \mathbf{0}, \tag{1}$$

where $\mathbf{D}_i \in \{\mathbf{D}_x, \mathbf{D}_y, \mathbf{D}_{xx}, \mathbf{D}_{yy}, \mathbf{D}_{xy}\}$ denote the spatial vertical and horizontal derivatives. The ℓ_1-norm of $\mathbf{E}$ encourages sparsity of the PSF entries. Additionally, another non-negativity constraint is imposed on the entries of $\mathbf{E}$. A wide range of PSFs for each DP view is estimated independently. Fig. S2 shows examples of the estimated DP- PSFs.

S2. Parameter search for our dual-pixel PSFs

In Sec. 2.2 and 5 of the main paper, we also mentioned a mechanism to select the effective range of parameters (i.e., n, α, β, κ) for our parametric DP-PSF modeling. Recall that n is the Butterworth filter order, and α is used to control its 3db cutoff position. β is the filter lower bound scale, and κ is the Gaussian smoothing factor.

Our goal is not to exactly match the measured PSFs found in cameras but rather to find representative PSFs that are very similar to what is estimated. The main reason is that we used a single camera, and as shown in Fig. S2, the estimated PSFs are noisy and not perfectly circular. This observation is expected due to camera-specific physical constraints like the positioning of the microlens, the depth of the sensor wells, and other optics manufacturing imperfections. Therefore, we limit the parameter search to a range of discrete values for each parameter sampled as:

$$\begin{aligned} n &\in \{1, 2, 3, \ldots, 15\}, , \\ \alpha, \beta &\in \{0.1, 0.2, \ldots, 1.0\}, \\ \kappa &\in \{0.14, 0.21, \ldots, 0.42\}. \end{aligned} \tag{2}$$

Then, we perform a brute-force search in this bounded space by solving the following equation:

$$\underset{n,\alpha,\beta,\kappa}{\arg\min} \sum_i \left\| \mathbf{D}_i(\mathbf{E} - \mathbf{H}(n, \alpha, \beta, \kappa)) \right\|_2^2, \tag{3}$$

where $\mathbf{E}$ is the estimated PSF from a real camera and $\mathbf{H}(n, \alpha, \beta, \kappa)$ is our parameterized PSF. We found the parameters achieve the highest similarity at:

$$\begin{aligned} n &\in \{3, 6, 9\}, \\ \alpha &\in \{0.4, 0.6, 0.8, 1.0\}, \\ \beta &\in \{0.1, 0.2, 0.3, 0.4\}, \\ \kappa &= 0.14. \end{aligned} \tag{4}$$

Given the best values we defined for each parameter, we can generate 48 combinations of possible PSFs, and those represent our bank of DP-PSFs used to generate our synthetic DP data. Fig. S2 shows examples of our parameterized PSFs in comparison with the estimated ones. Our PSFs demonstrate a much higher correlation with the estimated real PSFs compared to the DP-PSF model in [9]. The similarity is measured using the 2D cross-correlation $\mathcal{X}(.)$.

Such comprehensive PSF calibration (Sec. S1) and parameter search (Sec. S2) is not possible for smartphone cameras due to uncontrollable camera factors like aperture size and focal length. Additionally, up to our knowledge, only Pixel 3 and 4 smartphones allow direct access to the DP data, and the Pixel-DP API provided in [3] does not facilitate manual focus, which limits us from controlling the focus distance as well. The work in [9] estimated two PSFs from a Pixel 3 smartphone, and we found that they achieve a high correlation with our parametric PSF model.

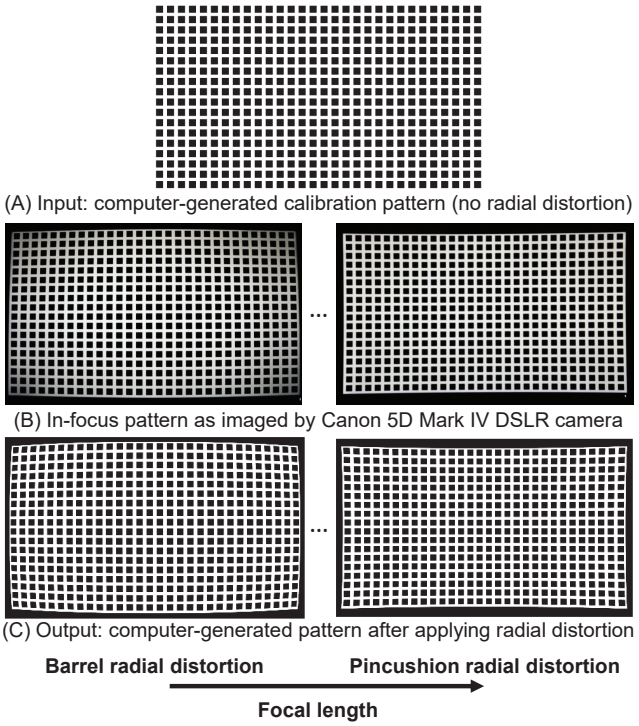


Figure S3. Calibration pattern used for estimating the radial distortion coefficients. A: the input computer-generated pattern with no radial distortion applied. B: the same pattern as imaged by the Canon 5D Mark IV camera at different focal lengths. C: the computer-generated pattern after applying radial distortion.

S3. Radial distortion coefficients

Based on Sec. 2.3 of the main paper, radial distortion is applied for more realistic imagery since the input images and our parametric DP-PSFs are not radially distorted. To this aim, we use the division model [2], as follows:

$$(x_d, y_d) = (x_o, y_o) + \frac{(x_u - x_o, y_u - y_o)}{1 + c_1 R^2 + c_2 R^4 + \cdots}, \tag{5}$$

where (x_u, y_u) and (x_d, y_d) are the undistorted and distorted pixel coordinates respectively, and c_i is the i^{th} radial distortion coefficient. R is the radial distance from the image plane center (x_o, y_o). This section introduces the calibration used to capture real-world radial distortion cases and is followed by the coefficient search procedure used to mimic real-world radial distortion. Recall that radial distortion is associated with zoom lenses and depends mainly on the camera's focal length.

We synthesize a uniform pattern of squares, as shown in Fig. S3-A. We follow the same setup described in Sec. S1. Still, with the following changes: (1) we capture an in-focus calibration pattern, (2) the focal length is changed across captures and the aperture remains fixed, (3) the distance between the display and camera is adjusted accordingly to make sure the full-resolution image is within cam-

era's field of view, sine increasing the focal length introduces zoom/magnification effect, and (4) the focus distance is also adjusted accordingly to make sure the pattern is in focus. Fig. S3-B shows examples of the calibration pattern as imaged by the Canon camera at different focal lengths.

We performed five captures at five different focal lengths ranging from min to max. Each is mapped to a focal length in our predefined parameter sets of the virtual five cameras – namely: $\{4, 5, 6\}, \{5, 8, 6\}, \{7, 5, 8\}, \{10, 13, 12\}, \{22, 10, 30\}$ — such that each set represents focal length, aperture size, and focus distance. With these five representative radial distortions that cover barrel as well as pincushion distortions, a brute-force search is performed to find the c_i coefficients that satisfy the following:

$$\arg\min_{c_1,c_2,c_3} \mathcal{X}(\mathbf{I}_f, \mathbf{I}_d(c_1, c_2, c_3)), \tag{6}$$

where $\mathbf{I}_f$ is the calibration pattern as imaged by the Canon camera at a certain focal length f (e.g., Fig. S3-B). $\mathbf{I}_d(c_1, c_2, c_3)$ is the computer-generated input pattern but after applying radial distortion based on the coefficients c_1, c_2, c_3 (see example in Fig. S3-C). Since we have few examples, $\mathbf{I}_f$ is re-centered manually to match $\mathbf{I}_d(c_1, c_2, c_3)$'s center. While Eq. 5 can be defined with more coefficients, we found three coefficients sufficient to approximate our real-world distortion examples. The final optimal five sets of coefficients are:

$$\begin{gathered}\{2 \times 10^{-2}, 2 \times 10^{-2}, 3 \times 10^{-2}\}, \\ \{8 \times 10^{-3}, 2 \times 10^{-3}, 2.2 \times 10^{-3}\}, \\ \{-4 \times 10^{-3}, 9 \times 10^{-4}, -9 \times 10^{-4}\}, \\ \{-7 \times 10^{-3}, -3.8 \times 10^{-3}, -3.6 \times 10^{-3}\}, \\ \{-8 \times 10^{-3}, -5 \times 10^{-3}, -4.5 \times 10^{-3}\}. \end{gathered} \tag{7}$$

S4. Ablation study

This section investigates the usefulness of our synthetically generated DP data along with our novel recurrent dual-pixel deblurring architecture (RDPD) — this is related to Sec. 4 and Sec. 5 of the main paper. To this aim, we divide the ablation study into two parts:

- Explore the effectiveness of our DP data generator components (Sec. S4.1), including: (1) our parametric DP-PSF model vs. the DP-PSF model presented in [9], (2) radially distorted DP data vs. undistorted ones, and (3) training with dual views vs. training with a single DP view.
- Investigate the effectiveness of each component added to our RDPD model (Sec. S4.2), including: (1) the utility of adding the radial patch distance mask to the input and (2) our new edge loss vs. traditional Sobel loss.

Table S1. Results on indoor and outdoor scenes combined from the Canon DP dataset [1]. Bold numbers are the best. RDPD+ (PSF [9]) is a variation trained on DP data generated using the PSF model from [9]. Our RDPD+, trained on DP data generated using our parametric DP-PSF, demonstrates +0.7db higher PSNR and reflects the power of our realistic PSF modeling.

Variation	PSNR ↑	SSIM ↑	MAE ↓
RDPD+ (PSF [9])	24.69	0.752	0.044
RDPD+ (our PSF)	**25.39**	**0.772**	**0.040**

Table S2. Results on indoor and outdoor scenes combined from the Canon DP dataset [1]. Bold numbers are the best. RDPD+ trained with radially distorted DP data achieves +0.2db higher PSNR when tested on real images, in which applying radial distortion on the synthetically generated DP data helped RDPD+ to learn the spatially varying PSFs shapes found in real cameras.

Variation	PSNR ↑	SSIM ↑	MAE ↓
RDPD+ (w/o distortion)	25.19	0.758	0.041
RDPD+ (w/ distortion)	**25.39**	**0.772**	**0.040**

Note that all subsequent experiments are conducted with variations of RDPD+. Each variation represents a single change, where the rest remains similar to RDPD+ as described in the main paper Sec. 5. All the variations are tested on Canon DP data from [1], since it is the only data that we can have access to real ground truth DP data and thus enables us to report quantitative results.

S4.1. Utility of DP data generator components

Our parametric DP-PSF. While our parametric DP-PSF model already has a higher correlation with the estimated PSFs from real cameras, we further investigate the effect on RDPD+ when trained with data generated using other DP-PSFs. In this study, we compare our RDPD+ that is trained with DP data generated using our parametric DP-PSF model against the DP data generated using the DP-PSF model in [9].

The quantitative results reported in Table S1 demonstrates the power of training RDPD+ with DP data that is generated using our realistically modeled PSFs, where there is an increase in PSNR of +0.7db.

Radial distortion. For more realistic modeling, we considered applying radial distortion in the proposed DP data generator. In this study, we examine the proposed deblurring RDPD+ model's behavior when it is also trained with data that is not radially distorted. In Table S2, we present the quantitative results of training RDPD+ with data generated with and without radial distortion. The results demonstrate the effectiveness of modeling the radial distortion during synthesizing the DP images, where RDPD+ trained with radially distorted data leads to a +0.2db PSNR gain.

Dual views vs. single view. Following [1], we explore the effect of training RDPD+ with DP views vs. training with

Table S3. Results on indoor and outdoor scenes combined from the Canon DP dataset [1]. Bold numbers are the best. RSPD+: recurrent single-pixel deblurring trained with a single DP view (i.e., left view). RDPD+: trained with left and right DP views. Utilizing DP views to train our RDPD+ leads to +1.15db PSNR gain and is essential for better defocus deblurring.

Variation	PSNR ↑	SSIM ↑	MAE ↓
RSPD+	24.24	0.726	0.045
RDPD+	**25.39**	**0.772**	**0.040**

Table S4. Results on indoor and outdoor scenes combined from the Canon DP dataset [1]. Bold numbers are the best. RDPD+ has an improved results when it is trained with the radial distance patch.

Variation	PSNR ↑	SSIM ↑	MAE ↓
RDPD+(w/o radial distance)	25.00	0.756	0.042
RDPD+(w/ radial distance)	**25.39**	**0.772**	**0.040**

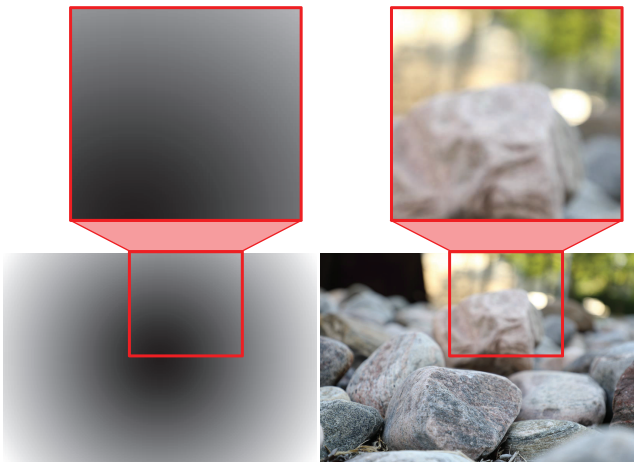


Figure S4. The radial distance patch used to assist RDPD+ training and address the issue of patch-wise training.

a single view (i.e., the traditional approach of single image deblurring [5, 6, 10]). To this aim, we introduce a recurrent single-pixel deblurring model variant (RSPD+) and compare it with our proposed RDPD+ model. Quantitative results in Table S3 demonstrates that training with DP views is crucial in order to perform better defocus deblurring where there is a +1.15db PSNR gain.

S4.2. Effectiveness of RDPD components

Radial distance patch for patch-wise training. We introduced training with the radial distance patch to feed each pixel's spatial location in the cropped patch and address the patch-wise training issue (i.e., the network does not see the full image or the relative position of the patch in the full image). Training in this manner is important as the PSFs are spatially varying in the radial direction away from the image center. Fig. S4 shows an example of the cropped radial distance patch used to assist our training. To investigate the effectiveness of training with the radial distance patch, we also train RDPD+ without it and report the results in Table S4. RDPD+ has a gain in PSNR of +0.4db when trained with the radial distance patch.

Table S5. Results on indoor and outdoor scenes combined from the Canon DP dataset [1]. Bold numbers are the best. **RDPD+(0)**: trained without the edge loss. **RDPD+(1)**: trained with a 3×3 single-scale Sobel loss similar to [7]. **RDPD+(3)**: trained with our three-scale edge loss. RDPD+ trained with our multi-scale edge loss has the best results for all metrics.

Variation	PSNR ↑	SSIM ↑	MAE ↓
RSPD+(0)	25.06	0.765	0.042
RSPD+(1) [7]	25.11	0.763	0.042
RDPD+(3)	**25.39**	**0.772**	**0.040**

Table S6. Results on indoor and outdoor scenes combined from the Canon DP dataset [1]. Bold numbers are the best. RDPD+ has about -0.5db PSNR drop in performance, when both our radial distance patch and multi-scale edge loss are removed from the RDPD+'s training.

Variation	PSNR ↑	SSIM ↑	MAE ↓
RDPD+(w/o both)	24.90	0.754	0.042
RDPD+(w/ both)	**25.39**	**0.772**	**0.040**

Our multi-scale edge loss. As mentioned in Sec. 4 of the main paper, we introduced the multi-scale edge loss based on the Sobel gradient operator to recover sharper details at different edge sizes. To examine our edge loss function's effectiveness, we train RDPD+ with different variations of edge loss scales denoted as RDPD+(m), where m is the number of scales used. In particular, we introduce RDPD+(0) (trained without the edge loss), RDPD+(1) (trained with a 3×3 single-scale Sobel loss similar to [7]), and RDPD+(3) (trained with our three-scale edge loss). Table S5 shows the quantitative results, in which training with our multi-scale edge loss achieves the best results for all metrics.

Effect of radial distance and edge loss together. We also examine the performance when our radial distance patch and multi-scale edge loss are removed from the RDPD+'s training. Table S6 shows the results, where there is a PSNR drop of -0.5db, indicating the usefulness of the additional proposed components.

S5. Defocus vs. motion deblurring

While defocus and motion both lead to image blur, the physical formation and appearance of these two blur types are significantly different. Therefore, methods that solve for motion blur are not expected to perform well when applied to defocus deblurring. This is shown by Abuolaim et al. [1], where the motion deblurring method of Tao et al. [11] that is evaluated on the same Canon test set achieves an average PSNR of 20.12dB, which is significantly lower than ours

(i.e., 25.39) and all other defocus deblurring methods.

S6. Additional qualitative results

As mentioned in Sec. 5 of the main paper, we provide more qualitative results for all existing defocus deblurring methods including a variations of ours. The methods are: the DP deblurring network (DPDNet) [1], the edge-based defocus blur (EBDB) [5], the defocus map estimation network (DMENet) [6], and the just noticeable blur (JNB) [10] estimation. Fig. S5, Fig. S6, and Fig. S7 show more qualitative results on images from the Canon dataset [1]. Fig. S8 shows results on images from a Pixel smartphone.

References

[1] Abdullah Abuolaim and Michael S Brown. Defocus deblurring using dual-pixel data. In *ECCV*, 2020. 4, 5, 6, 7, 8, 9, 10

[2] Andrew W Fitzgibbon. Simultaneous linear estimation of multiple view geometry and lens distortion. In *CVPR*, 2001. 3

[3] Rahul Garg, Neal Wadhwa, Sameer Ansari, and Jonathan T Barron. Learning single camera depth estimation using dual-pixels. In *ICCV*, 2019. 3

[4] Neel Joshi, Richard Szeliski, and David J Kriegman. Psf estimation using sharp edge prediction. In *CVPR*, 2008. 1

[5] Ali Karaali and Claudio Rosito Jung. Edge-based defocus blur estimation with adaptive scale selection. *TIP*, 27(3):1126–1137, 2017. 5, 6

[6] Junyong Lee, Sungkil Lee, Sunghyun Cho, and Seungyong Lee. Deep defocus map estimation using domain adaptation. In *CVPR*, 2019. 5, 6

[7] Zhengyang Lu and Ying Chen. Single image super resolution based on a modified u-net with mixed gradient loss. *arXiv preprint arXiv:1911.09428*, 2019. 5

[8] Fahim Mannan and Michael S Langer. Blur calibration for depth from defocus. In *Conference on Computer and Robot Vision (CRV)*, 2016. 1

[9] Abhijith Punnappurath, Abdullah Abuolaim, Mahmoud Afifi, and Michael S Brown. Modeling defocus-disparity in dual-pixel sensors. In *ICCP*, 2020. 1, 2, 3, 4

[10] Jianping Shi, Li Xu, and Jiaya Jia. Just noticeable defocus blur detection and estimation. In *CVPR*, 2015. 5, 6

[11] Xin Tao, Hongyun Gao, Xiaoyong Shen, Jue Wang, and Jiaya Jia. Scale-recurrent network for deep image deblurring. In *CVPR*, 2018. 5

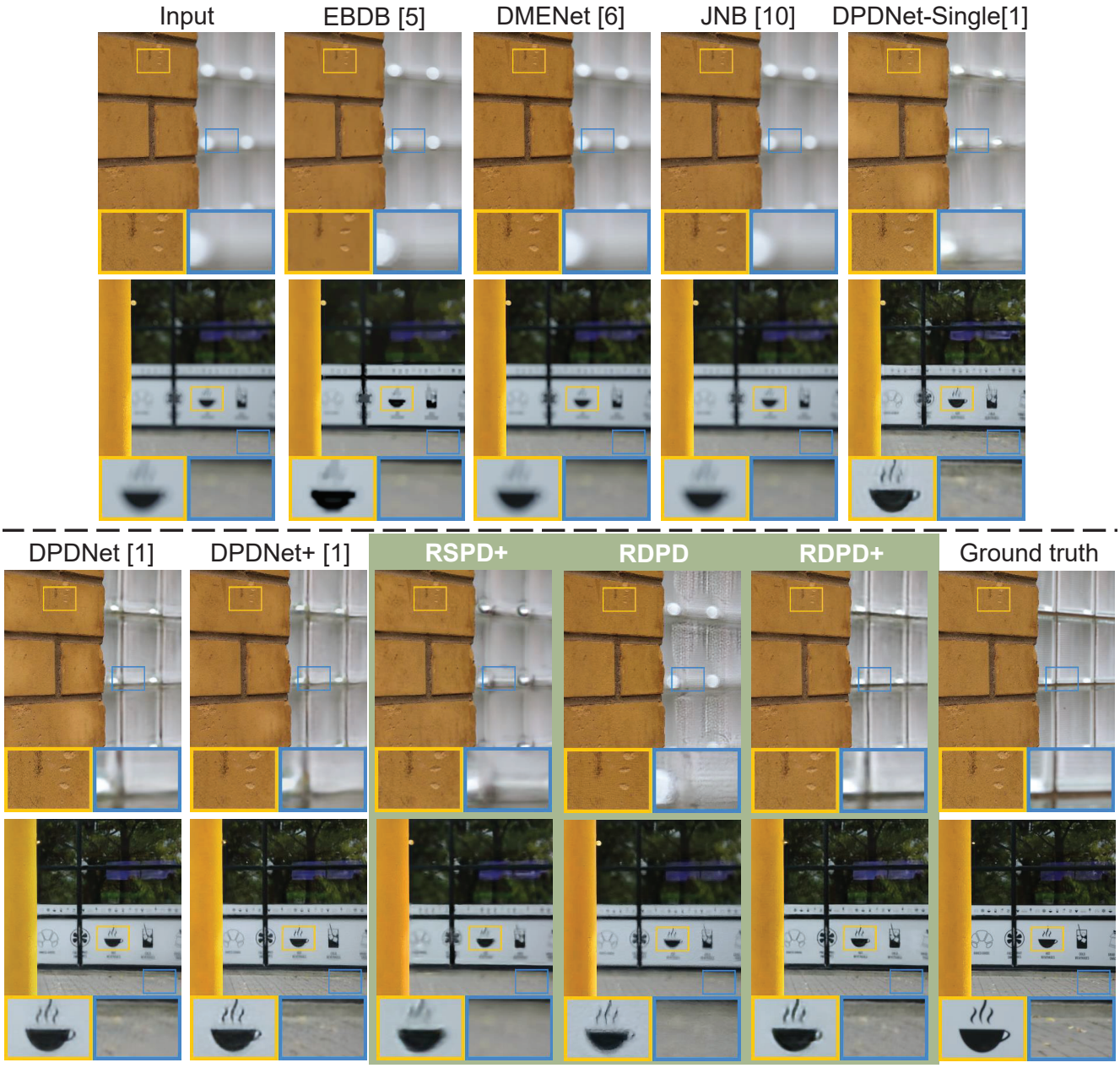


Figure S5. Qualitative results on images from Canon dataset [1]. DPDNet [1] is trained on Canon DP data. RDPD is our method trained on synthetically generated DP data only. DPDNet+ and RDPD+ are trained on both Canon and synthetic DP data. DPDNet-Single and RSPD+ are trained on a single DP view (i.e., $\mathbf{I}_l$). In general, RDPD and RDPD+ are able to recover more image details. Interestingly, RDPD trained on synthetic data generalizes well to real data from Canon camera.

Figure S6. Qualitative results on images from Canon dataset [1]. DPDNet [1] is trained on Canon DP data. RDPD is our method trained on synthetically generated DP data only. DPDNet+ and RDPD+ are trained on both Canon and synthetic DP data. DPDNet-Single and RSPD+ are trained on a single DP view (i.e., $\mathbf{I}_l$). In general, RDPD and RDPD+ are able to recover more image details. Interestingly, RDPD trained on synthetic data generalizes well to real data from Canon camera.

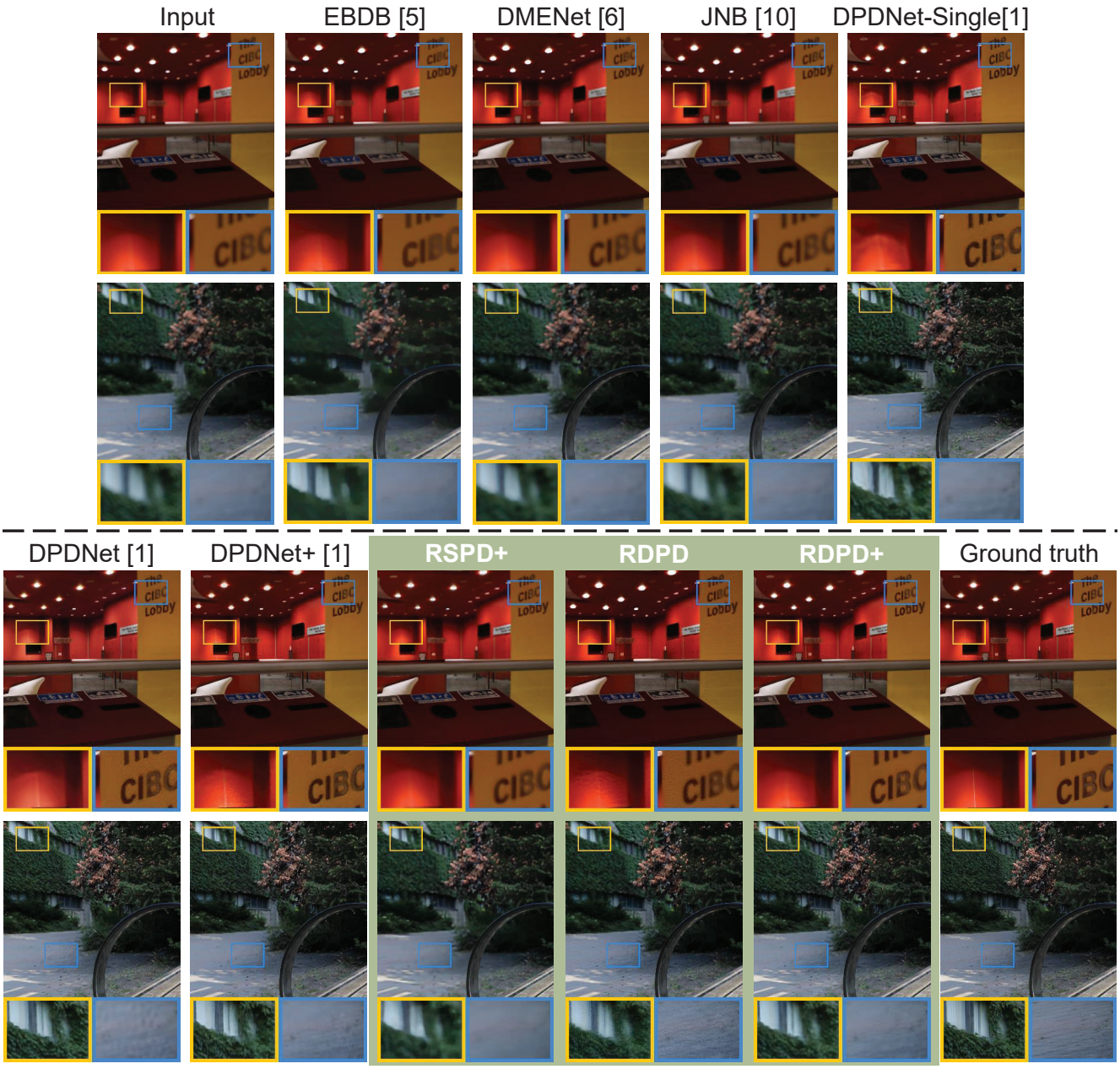


Figure S7. Qualitative results on images from Canon dataset [1]. DPDNet [1] is trained on Canon DP data. RDPD is our method trained on synthetically generated DP data only. DPDNet+ and RDPD+ are trained on both Canon and synthetic DP data. DPDNet-Single and RSPD+ are trained on a single DP view (i.e., $\mathbf{I}_l$). In general, RDPD and RDPD+ are able to recover more image details. Interestingly, RDPD trained on synthetic data generalizes well to real data from Canon camera.

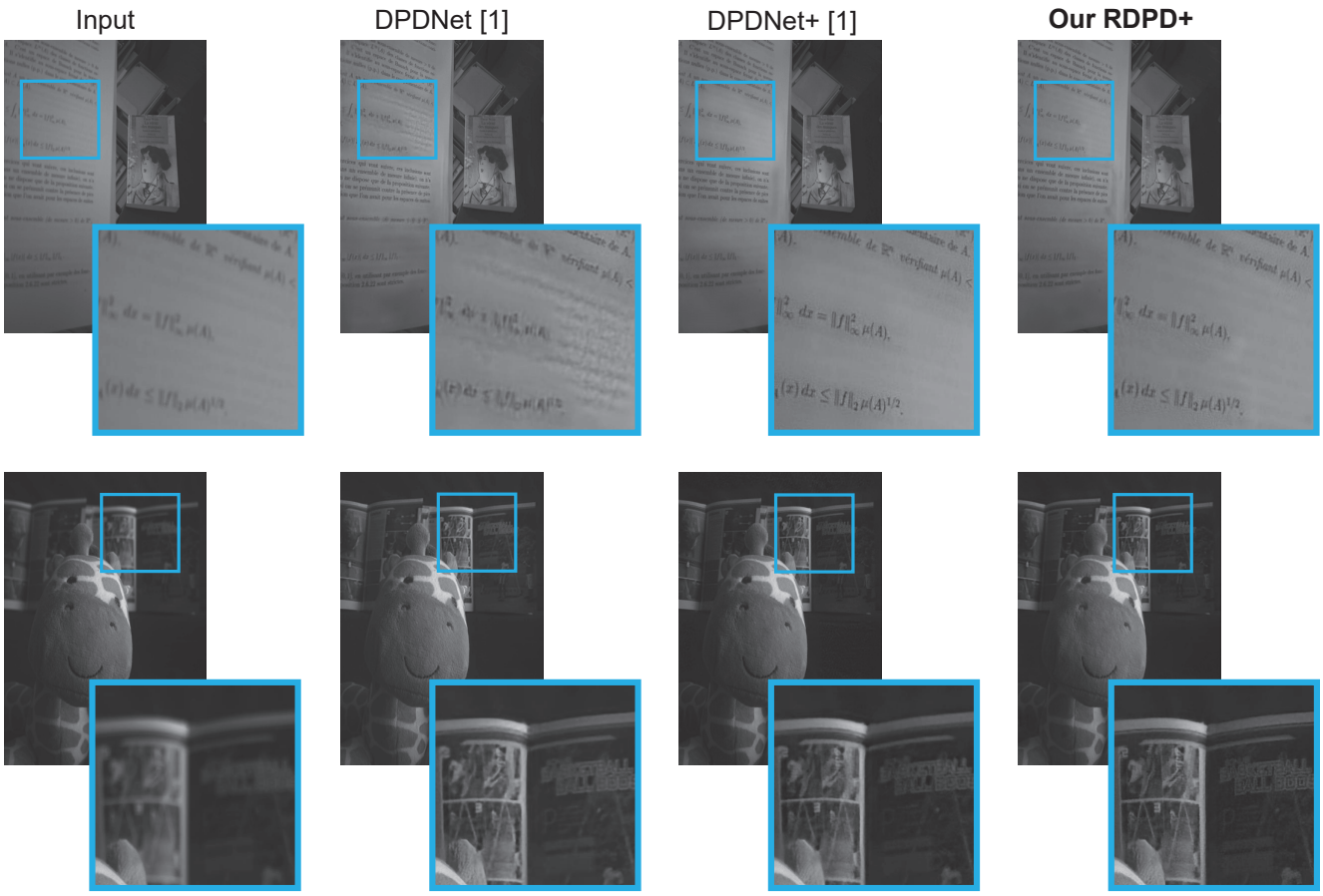


Figure S8. Qualitative results on images captured by Pixel smartphone. DPDNet [1] is trained on Canon DP data only. DPDNet+ and RDPD+ are trained on both Canon and synthetic DP data. In general, RDPD+ is able to recover more image details. Interestingly, DPDNet+ achieves better results when it is trained with our synthetic data augmented. Note that there is no ground truth sharp image for Pixel smartphone because smartphones have a fixed aperture. As a result, a narrow-aperture image cannot be captured to serve as a ground truth image. Additionally, we note that the DP data currently available from the Pixel smartphones are not full-frame but are limited to only one of the green channels in the raw-Bayer frame.