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

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

Abstract

Recent work has shown impressive results on data-driven defocus deblurring using the two-image views available on modern dual-pixel (DP) sensors. One significant challenge in this line of research is access to DP data. Despite many cameras having DP sensors, only a limited number provide access to the low-level DP sensor images. In addition, capturing training data for defocus deblurring involves a time-consuming and tedious setup requiring the camera's aperture to be adjusted. Some cameras with DP sensors (e.g., smartphones) do not have adjustable apertures, further limiting the ability to produce the necessary training data. We address the data capture bottleneck by proposing a procedure to generate realistic DP data synthetically. Our synthesis approach mimics the optical image formation found on DP sensors and can be applied to virtual scenes rendered with standard computer software. Leveraging these realistic synthetic DP images, we introduce a recurrent convolutional network (RCN) architecture that improves deblurring results and is suitable for use with single-frame and multi-frame data (e.g., video) captured by DP sensors. Finally, we show that our synthetic DP data is useful for training DNN models targeting video deblurring applications where access to DP data remains challenging.

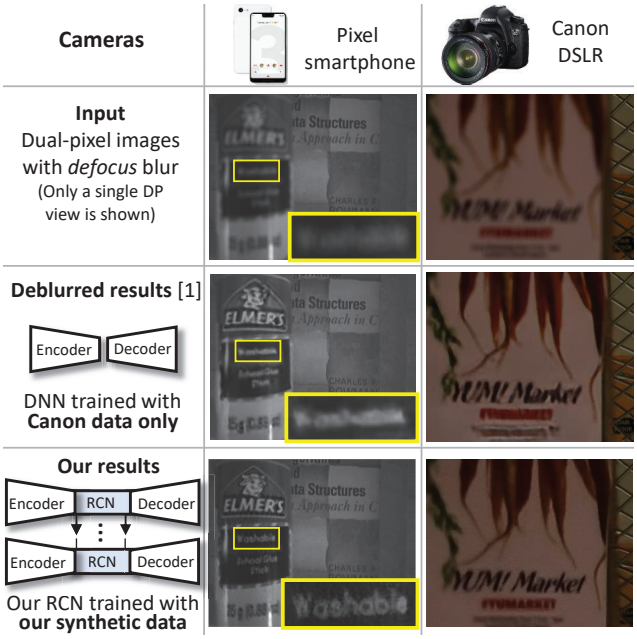

Figure 1. Deblurring results on images from a Pixel 4 smartphone and a Canon 5D Mark IV. **Third row:** results of the DNN proposed in [1] trained with DP data from the Canon camera. **Last row:** results from our proposed network trained on *synthetically generated data only*.

1. Introduction and related work

Defocus blur occurs in scene regions captured outside the camera's depth of field (DoF). Although the effect can be intentional (e.g., the bokeh effect in portrait photos), in many cases defocus blur is undesired as it impacts image quality due to the loss of sharpness of image detail (e.g., Fig. 1, second row). Recovering defocused image details is challenging due to the spatially varying nature of the defocus point spread function (PSF) [25, 41], which not only is scene depth dependent, but also varies based on the camera aperture, focal length, focus distance, radial distortion, and optical aberrations. Most existing deblurring methods [6, 20, 23, 31, 38] approach the defocus deblurring problem by first estimating a defocus image map. The defocus map is then used with an off-the-shelf non-blind deconvolution method (e.g., [7, 22]). This strategy to defocus deblurring is greatly limited by the accuracy of the estimated defocus map.

Recently, work in [1] was the first to propose an interesting approach to the defocus deblurring problem by leveraging information available on dual-pixel (DP) sensors found on most modern cameras. The DP sensor was originally designed to facilitate auto-focusing [2, 3, 19]; however, researchers have found DP sensors to be useful in broader applications, including depth map estimation [10, 30, 34, 47], defocus deblurring [4, 43, 24, 30], reflection removal [35], and synthetic DoF [44]. DP sensors consist of two photo-

*This work was done while Abdullah was an intern at Google.

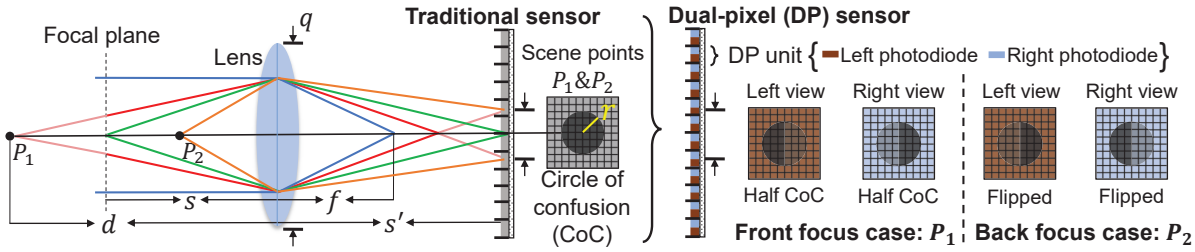

Figure 2. Thin lens model illustration and dual-pixel image formation. The circle of confusion (CoC) size is calculated for a given scene point using its distance from the lens, camera focal length, and aperture size. On the two dual-pixel views, the half-CoC PSF flips if the scene point is in front or back of the focal plane.

diodes at each pixel location effectively providing the functionality of a simple two-sample light-field camera (Fig. 2, DP sensor). Light rays coming from scene points within the camera's DoF will have no difference in phase, whereas light rays from scene points outside the camera's DoF will have a relative shift that is directly correlated to the amount of defocus blur. Recognizing this, the work in [1] proposed a deep neural network (DNN) framework to recover a deblurred image from a DP image pair using ground-truth data captured from a Canon DSLR.

Although the work of [1] demonstrates state-of-the-art deblurring results, it is restricted by the requirement for accurate ground truth data, which requires DP images to be captured in succession at different apertures. As well as being labor intensive, the process requires careful control to minimize the exposure and motion differences between captures (e.g., see local misalignment in Fig. 3). Another significant drawback is that data capture is limited to a *single* commercial camera, the Canon 5D Mark IV, the only device that currently provides access to raw DP data and has a controllable aperture.

While datasets exist for defocus estimation, including CUHK [37], DUT [48], and SYNDOF [23], as well as light-field datasets for defocus deblurring [6] and depth estimation [14, 40], none provide DP image views. The work of [1] is currently the only source of ground truth DP data suitable for defocus deblur applications but is limited to a single device. This lack of data is a severe limitation to continued research on data-driven DP-based defocus deblurring, in particular to applications where the collection of ground truth data is not possible (e.g., fixed aperture smartphones).

Contributions. This work aims to overcome the challenges in gathering ground-truth DP data for data-driven defocus deblurring. In particular, we propose a generalized model of the DP image acquisition process that allows realistic generation of synthetic DP data using standard computer graphics-generated imagery. We demonstrate that we can achieve state-of-the-art defocus deblurring results using synthetic data only (see Fig. 1), as well as complement real-image data sets through data augmentation. To demonstrate the generality of the model, we explore a new application domain of video defocus deblurring using DP data and propose a recurrent convolutional network (RCN) architecture that scales from single-image deblurring to video deblurring applications. Additionally, our proposed RCN addresses the issue of patch-wise training by incorporating radial distance learning and improves the deblurring results with a novel multi-scale edge loss. Our comprehensive experiments demonstrate the power of our synthetic DP data generation procedure and show that we can achieve the state-of-the-art results quantitatively and qualitatively with a novel network design trained with this data.

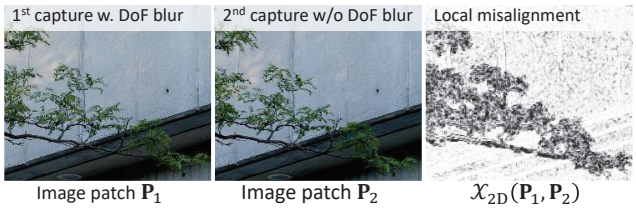

Figure 3. Misalignment in the Canon DP dataset [1] due to the physical capture procedure. Patches $\mathbf{P}_1$ and $\mathbf{P}_2$ are cropped from in-focus regions from two captures: the first using a wide-aperture (w/ defocus blur) and the second capture using a narrow-aperture (w/o defocus blur). The 2nd capture is intended to serve as the ground truth for this image pair. The third column shows the 2D cross correlation between the patches $\mathcal{X}_{2D}(\mathbf{P}_1, \mathbf{P}_2)$, which reveals the local misalignment that occurs in such data capture.

2. Modeling defocus blur in dual-pixel sensors

Synthetically generating realistic blur has been shown to improve data-driven approaches to both defocus map [23] and depth map estimation [29]. We follow a similar approach in this work but tackle the problem of generating realistic defocus blur targeting DP image sensors. For this, we comprehensively model the complete DP image acquisition process with spatially varying PSFs, radial lens distortion, and image noise. Fig. 4 shows an overview of our DP data generator that enables the generation of realistic DP views from an all-in-focus image and corresponding depth map.

2.1. Thin lens model

We model our virtual camera optics using a thin lens model that assumes negligible lens thickness, helping to simplify optical ray tracing calculations [32]. Through

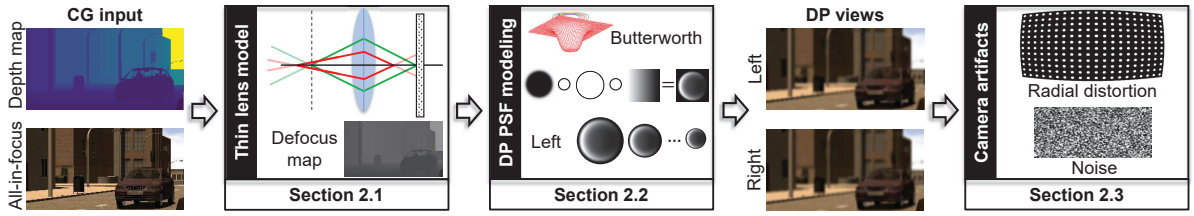

Figure 4. An overview of our framework used to synthetically generate dual-pixel (DP) views. Our approach starts with computer-generated (CG) imagery produced with a standard computer graphics package. Starting from this data, we model scene defocus, PSFs related to the DP sensor image formation, and additional artifacts, including radial distortion and sensor noise.

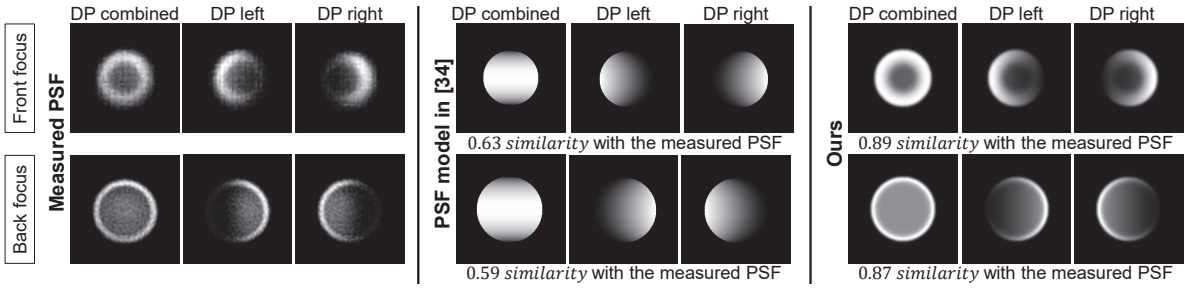

Figure 5. Front and back focus DP PSFs. The similarity between two PSFs is measured by the 2D cross correlation. **Left**: Measured DP PSFs from the Canon 5D Mark IV DSLR. **Middle**: DP PSFs as modeled by [34]. **Right**: Our newly proposed model for DP PSFs based on a modified 2D Butterworth filter. Our modeling achieves higher correlation with the real-world measured PSFs.

this model, we can approximate the circle of confusion (CoC) that represents the PSF for a given point based on its distance from the lens and camera parameters (i.e., focal length, aperture size, and focus distance). This model is illustrated in Fig. 2, where f is the focal length, s is the focus distance, and F is the f-stop. The distance between the lens and sensor s', and the aperture diameter q are defined as $s' = \frac{f\,s}{s-f}$ and $q = \frac{f}{F}$. Then, the CoC radius r of a scene point P_1 located at distance d from the camera is:

$$r = \frac{q}{2} \times \frac{s'}{s} \times \frac{d-s}{d}. \tag{1}$$

2.2. Dual-pixel PSFs

Recent work in [34] introduced a model for approximating a PSF that occurs in the *left* and *right* views of a DP sensor using a nice symmetry property between the *left* and *right* PSFs. However, the model involved a single free parameter only that was directly correlated to the CoC size (Fig. 5, middle column). Though the model is able to capture the symmetry property observed in a real DP PSF, the overall PSF did not sufficiently reflect the true structure exhibited by real-world PSFs, as illustrated in Fig. 5's left column. Real DP PSFs exhibit a donut-shaped depletion in the CoC that is attributed to optical aberrations [41].

To provide more realistic PSFs for the DP views, we introduce a parametric model based on the 2D Butterworth filter $\mathbf{B}$, defined as follows:

$$\mathbf{B}(x,y) = \left(1 + \left(\frac{D_o}{\sqrt{(x-x_o)^2+(y-y_o)^2}}\right)^{2n}\right)^{-1}, \tag{2}$$

where n is the filter order, and D_o is a parameter controlling the 3dB cutoff position. Aiming to capture the donut-shaped structure of the PSF, we define a parametric PSF model based on the Butterworth filter $\mathbf{B}$ as follows:

$$\mathbf{H} = \mathbf{B} \circ \mathbf{C}(x_o, y_o), \tag{3}$$

where $\mathbf{C}$ represents a circular disk with radius r matching the CoC radius as calculated in Eq. 1. The notation $\circ$ denotes the Hadamard product. Both $\mathbf{B}$ and $\mathbf{C}$ are centered at (x_o, y_o). D_o is a function of the radius r and is controlled by the parameter α. The values of $\mathbf{B}$ are re-scaled to $[\beta, 1]$, where the parameter $\beta > 0$ is introduced to control the minimum depletion at the kernel's center (which is always positive based on our observation of PSFs measured from real-world data). With our proposed model, the parameterized PSF $\mathbf{H}$ has a sharp fall-off about the circumference. Therefore, we smooth $\mathbf{H}$ by convolving it with a Gaussian kernel of standard deviation $\kappa \times r$, where $0 < \kappa << 1$.

Our modeling of $\mathbf{H}$ represents the combined DP PSF, which is formed as $\mathbf{H} = \mathbf{H}_l + \mathbf{H}_r$, where $\mathbf{H}_l$ and $\mathbf{H}_r$ are the *left* and *right* DP PSFs, respectively. Similar to the work in [34], we enforce the constraint of horizontal symmetry between $\mathbf{H}_l$ and $\mathbf{H}_r$, and express $\mathbf{H}_r$ as $\mathbf{H}_r = \mathbf{H}_l^f$, where $\mathbf{H}_l^f$ represents the *left* PSF flipped about the vertical axis. $\mathbf{H}_l$ can be shown as $\mathbf{H}$ with a gradual fall-off towards the right direction (see front-focus DP left in Fig. 5). Mathematically, we denote $\mathbf{H}_l$ as:

$$\mathbf{H}_l = \mathbf{H} \circ \mathbf{M}, \quad \text{s.t. } \mathbf{H}_l \geq \mathbf{0}, \text{ with} \sum \mathbf{H}_l = \tfrac{1}{2}, \tag{4}$$

where $\mathbf{M}$ is a 2D ramp mask with a constant decay. This

decay can be considered as an intensity fall-off (intensity/pixel) in a given direction. The direction is determined by the sign of the CoC radius calculated based on the thin lens model. The positive sign represents the front focus (i.e., blurring of objects behind the focal plane), whereas the negative sign represents the back focus (i.e., blurring of objects in front of the focal plane). Our PSF model, parameterized with five parameters, facilitates synthesizing PSF shapes more similar to what we measured in real cameras under different scenarios (see Fig. 5, right column). From this model, we can generate a bank of representative PSFs based on actual observations from real cameras. Additional details about the calibration procedure, PSF estimation method, and parameter searching are provided in the supplemental material.

2.3. Modeling additional camera artifacts

Radial lens distortion. Radial lens distortion occurs due to lens curvature imperfections causing straight lines in the real world to map to circular arcs in the image plane. This is a well-studied topic with many methods for modelling and correcting the radial distortion (e.g., [5, 8, 13, 33]). In our framework, we consider applying radial distortion to the synthetically generated images to mimic this effect found in real cameras. We adopt the widely used division model introduced in [8], as follows:

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

where (x_u, y_u) and (x_d, y_d) are the undistorted and distorted points 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 model enables different types of radial distortion, including barrel and pincushion. We generate representative radial distortion sets at different focal lengths found on cameras. A detailed description of this procedure is provided in the supplemental material.

Noise. Image noise is the undesirable occurrence of random variations in intensity or color information in images. Our initial input is CG-generated data that is noise-free. In order to synthesize realistic images, we add signal-dependent noise as the last step. We model the noise using a signal-dependent Gaussian distribution where the variance of noise is proportional to image intensity [9, 26]. Let $\mathbf{I}$ be the noiseless image and $\mathbf{N}$ a zero-mean Guassian noise layer; then our modeling of the signal-dependent Gaussian noise is $\mathbf{I}_{\text{noise}} = \mathbf{I} + \mathbf{I} \circ \mathbf{N}$, where $\mathbf{N} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \text{Id})$, and σ controls the noise strength.

3. Generating dual-pixel views

In this section we introduce the synthetic dataset used, followed by a description of the procedure to synthetically generate the DP *left* and *right* views. The source of our synthetic example data comes from the street view SYNTHIA dataset [16], which contains image sequences of photo-realistic GC-rendered images from a virtual city. Each sequence has 400 frames on average. The dataset contains a large diversity in scene setups, involving many objects, cities, seasons, weather conditions, day/night time, and so forth. The SYNTHIA dataset also includes the depth-buffer and labelled segmentation maps. In our framework, we use the depth map to apply synthetic defocus blur in the process of generating the DP views.

To blur an image based on the computed CoC radius r, we first decompose the image into discrete layers according to per-pixel depth values, where the maximum number of layers is set to 500. Then, we convolve each layer with our parameterized PSF, blurring both the image and mask of the depth layer. Next, we alpha-blend the blurred layer images in order of back-to-front, using the blurred masks as alpha values. For each all-in-focus video frame $\mathbf{I}_s$, we generate two images – namely, the *left* $\mathbf{I}_l$ and *right* $\mathbf{I}_r$ sub-aperture DP views – as follows (for simplicity, let $\mathbf{I}_s$ be a patch with all pixels from the same depth layer):

$$\mathbf{I}_l = \mathbf{I}_s * \mathbf{H}_l, \qquad \mathbf{I}_r = \mathbf{I}_s * \mathbf{H}_r, \qquad (6)$$

where $*$ denotes the convolution operation. Afterwards, the radial distortion is applied on $\mathbf{I}_l$, $\mathbf{I}_r$, and $\mathbf{I}_s$ based on the camera's focal length. Finally, we add signal-dependent noise layers (i.e., $\mathbf{N}_l$ and $\mathbf{N}_r$) for the two DP views that have the same σ, but are drawn independently. The final output defocus blurred image $\mathbf{I}_b$ is equal to $\mathbf{I}_l + \mathbf{I}_r$.

Our synthetically generated DP views exhibit a similar focus disparity to what we find in real data, where the in-focus regions show no disparity and the out-of-focus regions have defocus disparity.

4. Defocus deblurring image sequences

With the ability to generate synthetic DP data, we can shift our attention to training new RCN-based architectures addressing image sequences (e.g., video) captured with DP sensors. This is possible only by using our synthetic DP data, as no current device allows video DP data capture. As we will show, our method can be used for both image sequences and single-image inputs. In the context of image sequences, the amount of defocus blur changes based on the motion of the camera and scene's objects over time. In the presence of such motion, sample depth variation over a sequence of frames provides useful information for deblurring. Our work is the first to explore the domain of defocus deblurring on image sequences (e.g., video).

We adopt a data-driven approach for correcting defocus blur. We leverage a symmetric encoder-decoder CNN-based architecture with skip connections between corresponding feature maps [28, 36]. Skip connections are

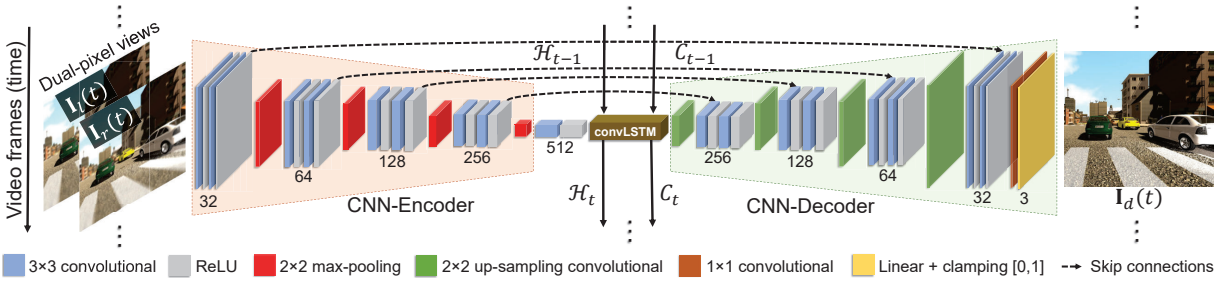

Figure 6. Our recurrent dual-pixel deblurring (RDPD) architecture. Our model takes a blurred image sequence, where each image at time t is fed as *left* $\mathbf{I}_l(t)$ and *right* $\mathbf{I}_r(t)$ DP views. The DP views are encoded at the encoder part to feed the convLSTM that outputs the hidden state $\mathcal{H}_t$ and memory cell $\mathcal{C}_t$ to the next time point. The convLSTM unit also outputs a feature map o_t that is processed through the decoder part to give the deblurred sharp image $\mathbf{I}_d(t)$. Note: the number of output filters is shown under each convolution operation.

widely used in encoder-decoder CNNs and have been found to be effective for image deblurring tasks [1, 11]. Our proposed network is also coupled with convLSTM units [42, 45, 46] to better learn temporal dependencies between multiple frames and to allow variable sequence size. With convLSTM units, the same network remains fully convolutional and can successfully deblur a single image or a sequence of images of arbitrary number. Fig. 6 shows a detailed overview of our proposed CNN-convLSTM architecture, which we refer to as *recurrent dual-pixel deblurring* (RDPD).

Our architecture is similar to the one in [1], but with the following modifications: (1) convLSTM units are added to the network bottleneck, (2) we train the network using the radial distance patch to address the patch-wise training issue, (3) we introduce a multi-scale edge loss function that helps in recovering sharp edges, (4) the number of nodes are reduced to half at each block to make the model lighter, and (5) the last layer is replaced by a linear layer with a [0,1] clamping as it is found to be more effective in [11].

RDPD architecture. Given an input video of j consecutive frames that have defocus blur $\{\{\mathbf{I}_l(t), \mathbf{I}_r(t)\}, \cdots, \{\mathbf{I}_l(t+j), \mathbf{I}_r(t+j)\}\}$ (such that $\mathbf{I}_l(t)$ and $\mathbf{I}_r(t)$ are the DP views of the given frame at time t), we first obtain a sequence of compact convolutional features $\{X(t), \cdots, X(t+j)\}$ encoded at the CNN bottleneck – namely, $X(t) = \text{CNN-Encoder}(\mathbf{I}_l(t), \mathbf{I}_r(t))$. Then, the features are fed to a convLSTM as shown in Fig. 6. We utilize the convLSTM to learn of the temporal dynamics of the sequential inputs. This is achieved by incorporating memory units with the gated operations. The convLSTM also preserves spatial information by replacing dot products with convolutional operations, which is essential for making spatially variant estimation align with the spatially varying DP PSFs. We choose LSTM over RNN because standard RNNs are known to have difficulty in learning long-time dependencies [17], whereas LSTMs have shown the capability to learn long- and short-time dependencies [18].

For the input feature $X(t)$ at time t, our convLSTM leverages three convolution gates – input i_t, output o_t, and forget $\mathcal{F}_t$ – in order to control the signal flow within the cell. The convLSTM outputs a hidden state $\mathcal{H}_t$ and maintains a memory cell $\mathcal{C}_t$ for controlling the state update and output:

$$i_t = \Sigma(W_i^X * X_t + W_i^{\mathcal{H}} * \mathcal{H}_{t-1} + W_i^{\mathcal{C}} \circ \mathcal{C}_{t-1} + b_i), \quad (7)$$

$$\mathcal{F}_t = \Sigma(W_{\mathcal{F}}^X * X_t + W_{\mathcal{F}}^{\mathcal{H}} * \mathcal{H}_{t-1} + W_{\mathcal{F}}^{\mathcal{C}} \circ \mathcal{C}_{t-1} + b_{\mathcal{F}}), \quad (8)$$

$$o_t = \Sigma(W_o^X * X_t + W_o^{\mathcal{H}} * \mathcal{H}_{t-1} + W_o^{\mathcal{C}} \circ \mathcal{C}_{t-1} + b_o), \quad (9)$$

$$\mathcal{C}_t = \mathcal{F}_t \circ \mathcal{C}_{t-1} + i_t \circ \tau(W_{\mathcal{C}}^X * X_t + W_{\mathcal{C}}^{\mathcal{H}} * \mathcal{H}_{t-1} + b_{\mathcal{C}}), \quad (10)$$

$$\mathcal{H}_t = o_t \circ \tau(\mathcal{C}_t), \quad (11)$$

where the W terms denote the different weight matrices, and the b terms represent the different bias vectors. Σ and τ are the activation functions of logistic sigmoid and hyperbolic tangent, respectively. Afterwords, the output deblurred image $\mathbf{I}_d$ is obtained by decoding o_t through the decoder part of our encoder-decoder CNN as follows:

$$\mathbf{I}_d(t) = \text{CNN-Decoder}(o_t). \quad (12)$$

Radial distance patch. Radial distortion and lens aberration make the PSFs vary in radial directions away from the image center. Similar to [1, 31], we perform patch-wise training to avoid the redundancies of full image training and ensure that the input has enough variance. However, this approach breaks the spatial correlation between the image patches as they are fed independently with no knowledge of their location on the image plane. As a result, in addition to the six-channel RGB DP views, we include a single-channel patch that represents the relative radial distance.

Multi-scale edge loss. In addition to the MSE loss, we introduce a multi-scale edge loss using a Sobel gradient to guide the network to encourage sharper edges. Our new loss is similar in principle to the single-scale (i.e., 3×3) Sobel loss used in [27], but we modified this loss in two ways: first, we added multiple scales of the Sobel operator (i.e., kernel sizes) in order to capture different edge sizes. Second, we minimized for the horizontal and vertical directions separately, to concentrate more on the direction that is perpendicular to the imaging sensor orientation. For our

Table 1. Results on the Canon DP dataset from [1]. DPDNet is the pre-trained model on Canon data provided by [1]. DPDNet+ and our RDPD+ are trained with Canon and our synthetically generated DP data. Bold numbers are the best and highlighted in green. The second-best performing results are highlighted in yellow. The testing set consists of 37 indoor and 39 outdoor scenes.

Method	Indoor			Outdoor			Indoor & Outdoor				Time ↓
	PSNR ↑	SSIM ↑	MAE ↓	PSNR ↑	SSIM ↑	MAE ↓	PSNR ↑	SSIM ↑	MAE ↓	NIQE ↓	
EBDB [20]	25.77	0.772	0.040	21.25	0.599	0.058	23.45	0.683	0.049	5.42	929.7
DMENet [23]	25.70	0.789	0.036	21.51	0.655	0.061	23.55	0.720	0.049	4.85	613.7
JNB [38]	26.73	0.828	0.031	21.10	0.608	0.064	23.84	0.715	0.048	5.11	843.1
DPDNet [1]	27.48	0.849	0.029	**22.90**	**0.726**	**0.052**	25.13	**0.786**	0.041	3.77	0.5
DPDNet+ [1]	27.65	**0.852**	0.028	22.72	0.719	0.054	25.12	0.784	0.042	3.73	0.5
Our RDPD+	**28.10**	0.843	**0.027**	22.82	0.704	0.053	**25.39**	0.772	**0.040**	**3.19**	**0.3**

multi-scale modified edge loss, the vertical G^x and horizontal G^y derivative approximations of the deblurred output I_d and its ground truth I_s are:

$$G_d^x = \mathrm{I}_d * S_{m\times m}^x, \qquad G_d^y = \mathrm{I}_d * S_{m\times m}^y, \tag{13}$$

$$G_s^x = \mathrm{I}_s * S_{m\times m}^x, \qquad G_s^y = \mathrm{I}_s * S_{m\times m}^y, \tag{14}$$

where $S_{m\times m}^x$ and $S_{m\times m}^y$ are the vertical and horizontal Sobel operators of size m, respectively. The derivative operations are performed at multiple filter sizes. Our new edge loss $\mathcal{L}_{\text{edge}}$ is the mean of multiple scales for each direction x/y and denoted as:

$$\mathcal{L}_{\text{edge}}^{\{x,y\}} = \mathbb{E}[\text{MSE}(G_s^{\{x,y\}}, G_d^{\{x,y\}})]. \tag{15}$$

Then the final loss function $\mathcal{L}$ is:

$$\mathcal{L} = \mathcal{L}_{\text{MSE}} + \lambda_x \mathcal{L}_{\text{edge}}^x + \lambda_y \mathcal{L}_{\text{edge}}^y, \tag{16}$$

such that $\mathcal{L}_{\text{MSE}}$ is the typical MSE loss between the output estimated I_d and its ground truth I_s. The λ terms are added to control our final loss.

5. Experiments

We evaluate our proposed RDPD and other existing defocus deblurring methods: the DP deblurring network (DPDNet) [1], the edge-based defocus blur (EBDB) [20], the defocus map estimation network (DMENet) [23], and the just noticeable blur (JNB) [38] estimation. DPDNet [1] is the only method that utilizes DP data for deblurring, and the others [20, 23, 38] use only a single image as input (i.e., I_l) and estimate the defocus map in order to feed it to an off-the-shelf deconvolution method (i.e., [7, 22]). EBDB [20] and JNB [38] are not learning-based methods; thus, we can test them directly. For the learning-based DMENet method, we cannot retrain it with the Canon data [1], as it does not provide the ground truth defocus map. However, with our data generator we are able to generate defocus maps, which allows us to retrain DMENet with our synthetically generated data.

Settings to generate DP data. For our DP data generator, we define five camera parameter sets – 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. Given the depth range found in the SYNTHIA dataset [16], these camera sets cover a wide range of front- as well as back-focus CoC sizes. For each image sequence in the SYNTHIA dataset, we generate five sequences based on the predefined camera sets. The radial distortion coefficients are set accordingly for each camera set. For the DP PSFs, we generate many representative PSF shapes by varying the parameters in the given ranges $n \in \{3,6,9\}$, $\alpha \in \{0.4,0.6,0.8,1\}$, $\beta \in \{0.1,0.2,0.3,0.4\}$, and $\kappa = 0.14$. Image noise layer strength is chosen randomly, where $\sigma \in \{5e^{-2}, 5.5e^{-2}, \cdots, 5e^{-1}\}$. These parameters are set empirically to model real camera hardware. More detail is provided in the supplemental material.

We divide the SYNTHIA dataset [16] into training and testing sequences. We generate five sets of blurred images for each image sequence. In total, we synthesize 2023 training and 201 testing blurred DP views. Though our synthetic DP data generator enables an unlimited number of images to be generated, we found this number of images sufficient for training. In addition to our synthetically generated DP data, we use the DP ground truth data from [1] with 300 training, 74 validation, and 76 testing pairs of blurred images (with DP views) and corresponding sharp images.

RDPD settings and training procedure. We set the size of the convLSTM to 512 units. For patch-wise training, we fix the size of input and output layers to $512 \times 512 \times 7$ and $512 \times 512 \times 3$, respectively. We initialize the weights of the convolutional layers using He's initialization [15] and use the Adam optimizer [21] to train the model. The initial learning rate is 5×10^{-5}, which is decreased by half every 40 epochs.

For domain generalization from synthetic to real data [12, 39], we train our model iteratively using mini-batches of real (i.e., single image) as well as synthetic data (i.e., image sequence), where the patches are randomly cropped at each iteration. This type of iterative image/image sequence training becomes feasible since our recurrent model RDPD allows training and testing with any

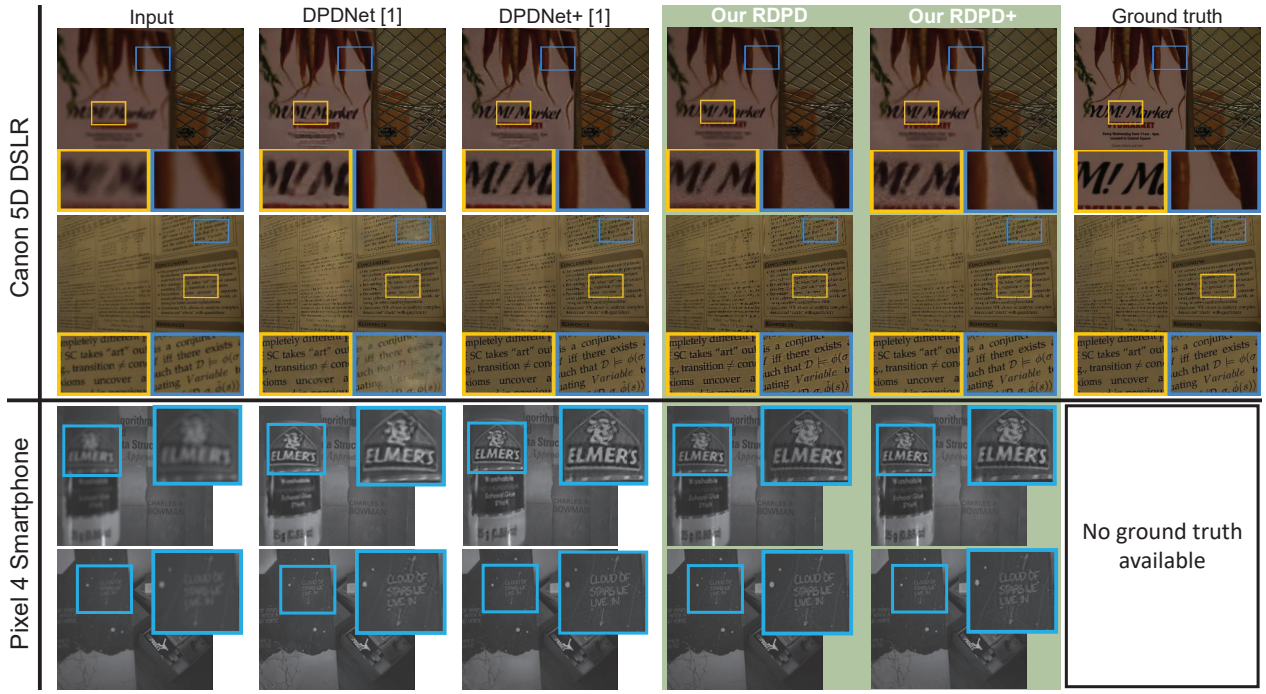

Figure 7. Qualitative results. DPDNet [1] is trained on Canon DP data. RDPD is our method trained on synthetically generated DP data only. DPDNet+ and RDPD+ are trained on *both* Canon and synthetic DP data. In general, RDPD and RDPD+ are able to recover more image details. Interestingly, RDPD trained on synthetic data generalizes well to real data from the two tested cameras. Note that there is no ground truth sharp image for Pixel 4, due to the fact smartphones have fixed aperture and thus 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.

number of frames, and it does not need to be preset beforehand. We set the mini-batch size for the real data iteration to eight batches, because the dataset of real data has only single-image examples (i.e., no image sequences). For the synthetic data iteration, we set the mini-batch size to two sequences each of size four frames. We define three scales for our edge loss – namely, $m \in \{3, 7, 11\}$. The λ terms are found to be effective at $\lambda_x = 0.03$ and $\lambda_y = 0.02$.

To avoid overfitting, the dropout layer in the convLSTM is set to 0.4. Our model converges after 140 epochs. Although we train on image patches, our RDPD (with convLSTM) is fully convolutional and enables testing on full-resolution inputs. To demonstrate the effectiveness of each component in our model, an ablation study of different training settings is provided in the supplemental material.

Single image results. We evaluate our proposed RDPD against existing defocus deblurring methods for single-image inputs. For methods that utilize DP views for the input image (i.e., RDPD and DPDNet [1]), we introduce variations on the training data used for more comprehensive evaluations. The variations are RDPD+ and DPDNet+ that are trained on both Canon DP data from [1] combined with synthetic DP data generated by the process described in Sec. 2. The RDPD without the + sign is our baseline trained with synthetically generated DP data only. The DPDNet without the + is trained on Canon data only.

In Table 1, we report quantitative results on real Canon DP data from [1] using standard metrics – namely, MAE, PSNR, SSIM, and time. We also report the Naturalness Image Quality (NIQE) metric of the output deblurred images with respect to a reference model derived from the DP GT images. In general, our RDPD+ has the best overall PSNR compared to other methods. Particularly, RDPD+ achieves the best PSNR and MAE for both indoor and combined categories, and all with our lighter-weight network that enabled the fastest inference time.

For the Outdoor dataset, the PSNR of RDPD+ is slightly lower (i.e., 0.08dB) due to the fact that the Outdoor dataset is imperfect as a result of the capturing process (see Fig. 3). DP cameras do not enable simultaneous capture of the DP images and corresponding ground truth sharp image (i.e., the image pairs can be captured only in succession at different times). As a result, the Outdoor ground-truth is imperfect with small local motion and illumination variations. The Indoor ground-truth is captured in more controlled conditions and has fewer imperfections. The slightly better performance of DPDNet for outdoor scenes is because DPDNet is learning to compensate for imperfections in the Outdoor dataset. A key strength of our work is the ability to synthetically generate DP data that is not impeded by

Figure 8. Results on a Canon 5D DSLR image sequence. PSNR is shown for each deblurred image. sRDPD+ has a 0.4dB lower PSNR on average when it is trained with a single frame compared to our multi-frame method – that is, RDPD+.

imperfections of manual capture. RDPD+ is trained on the Outdoor dataset as well as the synthetically generated data (without such imperfections) debiasing the result from the imperfect ground-truth. The consequence is reduced fidelity to the imperfect ground-truth (in terms of PSNR/SSIM), but better defocus deblur performance overall. Similar behavior is observed when DPDNet is trained with Canon data and our synthetically generated data (i.e., DPDNet+).

In Fig. 7, we also provide qualitative results of RDPD compared to other methods on data captured by Canon DSLR and Pixel 4 cameras. In general, RDPD+ is able to recover more details from the input deblurred image. Additionally, Fig. 7 demonstrates that the baseline RDPD achieves good deblurring results on Canon and Pixel 4 data despite being trained with synthetic data only. This result demonstrates the accuracy of the proposed framework for synthetic DP data generation and the ability of the recurrent model to generalize to different cameras. It can also be seen that DPDNet+ has improved results compared to DPDNet, demonstrating the benefit gained by DPDNet+ through the addition of synthetic DP data on training. The supplemental material contains more quantitative results, visual comparisons, and animated deblurring examples for both Canon and Pixel 4 cameras.

Image sequence results. Our RDPD is designed to handle input image sequences. Here, we investigate the improvement gained by training with image sequences vs. single frames. For this comparison, we introduce the RDPD+ variant sRDPD+, which is trained with single-frame inputs.

Table 2. Results on our synthetically generated DP data. sRDPD+ is a variation that is trained with single-frame data (**green**=best, **yellow**=second best). Our RDPD+, trained with image sequences, achieves the best results.

Method	PSNR ↑	SSIM ↑	MAE ↓
DPDNet [1]	26.38	0.782	0.034
DPDNet+ [1]	29.84	0.828	0.025
sRDPD+	30.26	0.849	0.020
RDPD+	**31.09**	**0.861**	**0.016**

As previously mentioned, there is no camera that enables access to DP views for video data. Nevertheless, we mimic the same capturing procedure in [1] in order to capture a sequence of images. We performed four captures of the same scene with small camera motion introduced between the captures. Each image has its own DP views and is captured at narrow and wide apertures. Fig. 8 presents the results on the sequence of images. The effectiveness of training with image sequences with RDPD+ can be seen from the average PSNR gain (i.e., +0.4dB) compared to sRDPD+ trained using single-image inputs.

Table 2 shows the quantitative results on our synthetically generated DP image sequences. Our method RDPD+ (trained on multiple frames) achieves the best results as it utilizes the convLSTM architecture to better model the temporal dependencies in an image sequence. Recall that our RDPD network is lighter and has a much lower number of weights compared to DPDNet.

6. Conclusion

We proposed a novel framework to generate realistic DP data by modeling the image formation steps present on cameras with DP sensors. Our framework helps to address the current challenges in capturing DP data. Utilizing our synthetic DP data, we also proposed a new recurrent convolutional architecture that is designed to reduce defocus blur in image sequences. We performed a comprehensive evaluation of existing deblurring methods, and demonstrated that our synthetically generated DP data and recurrent convolutional model achieve state-of-the-art results quantitatively and qualitatively. Furthermore, our proposed framework demonstrates the ability to generalize across different cameras by training on synthetic data only. We believe our DP data generator will help spur additional ideas about defocus deblurring and applications that leverage DP data. Our dataset, code, and trained models are available at https://github.com/Abdullah-Abuolaim/recurrent-defocus-deblurring-synth-dual-pixel.

Acknowledgments. The authors would like to thank Shumian Xin, Yinxiao Li, Neal Wadhwa, and Rahul Garg for fruitful comments and discussions.

References

[1] Abdullah Abuolaim and Michael S Brown. Defocus deblurring using dual-pixel data. In *ECCV*, 2020. 1, 2, 5, 6, 7, 8

[2] Abdullah Abuolaim and Michael S Brown. Online lens motion smoothing for video autofocus. In *WACV*, 2020. 1

[3] Abdullah Abuolaim, Abhijith Punnappurath, and Michael S Brown. Revisiting autofocus for smartphone cameras. In *ECCV*, 2018. 1

[4] Abdullah Abuolaim, Radu Timofte, Michael S Brown, et al. NTIRE 2021 challenge for defocus deblurring using dual-pixel images: Methods and results. In *CVPR Workshops*, 2021. 1

[5] Faisal Bukhari and Matthew N Dailey. Automatic radial distortion estimation from a single image. *Journal of mathematical imaging and vision*, 45(1):31–45, 2013. 4

[6] Laurent D'Andrès, Jordi Salvador, Axel Kochale, and Sabine Süsstrunk. Non-parametric blur map regression for depth of field extension. *TIP*, 25(4):1660–1673, 2016. 1, 2

[7] DA Fish, AM Brinicombe, ER Pike, and JG Walker. Blind deconvolution by means of the richardson–lucy algorithm. *Journal of the Optical Society of America (A)*, 12(1):58–65, 1995. 1, 6

[8] Andrew W Fitzgibbon. Simultaneous linear estimation of multiple view geometry and lens distortion. In *CVPR*, 2001. 4

[9] Alessandro Foi, Mejdi Trimeche, Vladimir Katkovnik, and Karen Egiazarian. Practical poissonian-gaussian noise modeling and fitting for single-image raw-data. *TIP*, 17(10):1737–1754, 2008. 4

[10] Rahul Garg, Neal Wadhwa, Sameer Ansari, and Jonathan T Barron. Learning single camera depth estimation using dual-pixels. In *ICCV*, 2019. 1

[11] Jochen Gast and Stefan Roth. Deep video deblurring: The devil is in the details. In *ICCV Workshops*, 2019. 5

[12] Xiaoyang Guo, Hongsheng Li, Shuai Yi, Jimmy Ren, and Xiaogang Wang. Learning monocular depth by distilling cross-domain stereo networks. In *ECCV*, 2018. 6

[13] Richard Hartley and Sing Bing Kang. Parameter-free radial distortion correction with center of distortion estimation. *TPAMI*, 29(8):1309–1321, 2007. 4

[14] Caner Hazirbas, Sebastian Georg Soyer, Maximilian Christian Staab, Laura Leal-Taixé, and Daniel Cremers. Deep depth from focus. In *ACCV*, 2018. 2

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 2015. 6

[16] Daniel Hernandez-Juarez, Lukas Schneider, Antonio Espinosa, David Vazquez, Antonio M. Lopez, Uwe Franke, Marc Pollefeys, and Juan Carlos Moure. Slanted stixels: Representing san francisco's steepest streets. In *BMVC*, 2017. 4, 6

[17] Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, Jürgen Schmidhuber, et al. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, 2001. 5

[18] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 5

[19] Jinbeum Jang, Yoonjong Yoo, Jongheon Kim, and Joonki Paik. Sensor-based auto-focusing system using multi-scale feature extraction and phase correlation matching. *Sensors*, 15(3):5747–5762, 2015. 1

[20] Ali Karaali and Claudio Rosito Jung. Edge-based defocus blur estimation with adaptive scale selection. *TIP*, 27(3):1126–1137, 2017. 1, 6

[21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[22] Dilip Krishnan and Rob Fergus. Fast image deconvolution using hyper-laplacian priors. In *NeurIPS*, 2009. 1, 6

[23] Junyong Lee, Sungkil Lee, Sunghyun Cho, and Seungyong Lee. Deep defocus map estimation using domain adaptation. In *CVPR*, 2019. 1, 2, 6

[24] Junyong Lee, Hyeongseok Son, Jaesung Rim, Sunghyun Cho, and Seungyong Lee. Iterative filter adaptive network for single image defocus deblurring. In *CVPR*, 2021. 1

[25] Anat Levin, Yair Weiss, Fredo Durand, and William T Freeman. Understanding blind deconvolution algorithms. *TPAMI*, 33(12):2354–2367, 2011. 1

[26] Ce Liu, Richard Szeliski, Sing Bing Kang, C Lawrence Zitnick, and William T Freeman. Automatic estimation and removal of noise from a single image. *TPAMI*, 30(2):299–314, 2007. 4

[27] 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

[28] Xiaojiao Mao, Chunhua Shen, and Yu-Bin Yang. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In *NeurIPS*, 2016. 4

[29] Maxim Maximov, Kevin Galim, and Laura Leal-Taixé. Focus on defocus: bridging the synthetic to real domain gap for depth estimation. In *CVPR*, 2020. 2

[30] Liyuan Pan, Shah Chowdhury, Richard Hartley, Miaomiao Liu, Hongguang Zhang, and Hongdong Li. Dual pixel exploration: Simultaneous depth estimation and image restoration. In *CVPR*, 2021. 1

[31] Jinsun Park, Yu-Wing Tai, Donghyeon Cho, and In So Kweon. A unified approach of multi-scale deep and hand-crafted features for defocus estimation. In *CVPR*, 2017. 1, 5

[32] Michael Potmesil and Indranil Chakravarty. A lens and aperture camera model for synthetic image generation. *SIGGRAPH*, 15(3):297–305, 1981. 2

[33] B Prescott and GF McLean. Line-based correction of radial lens distortion. *Graphical Models and Image Processing*, 59(1):39–47, 1997. 4

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

[35] Abhijith Punnappurath and Michael S Brown. Reflection removal using a dual-pixel sensor. In *CVPR*, 2019. 1

[36] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 4

[37] Jianping Shi, Li Xu, and Jiaya Jia. Discriminative blur detection features. In *CVPR*, 2014. 2

[38] Jianping Shi, Li Xu, and Jiaya Jia. Just noticeable defocus blur detection and estimation. In *CVPR*, 2015. 1, 6

[39] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Joshua Susskind, Wenda Wang, and Russell Webb. Learning from simulated and unsupervised images through adversarial training. In *CVPR*, 2017. 6

[40] Pratul P Srinivasan, Tongzhou Wang, Ashwin Sreelal, Ravi Ramamoorthi, and Ren Ng. Learning to synthesize a 4D RGBD light field from a single image. In *ICCV*, 2017. 2

[41] Huixuan Tang and Kiriakos N Kutulakos. Utilizing optical aberrations for extended-depth-of-field panoramas. In *ACCV*, 2012. 1, 3

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

[43] Tu Vo. Attention! stay focus! In *CVPR Workshops*, 2021. 1

[44] Neal Wadhwa, Rahul Garg, David E Jacobs, Bryan E Feldman, Nori Kanazawa, Robert Carroll, Yair Movshovitz-Attias, Jonathan T Barron, Yael Pritch, and Marc Levoy. Synthetic depth-of-field with a single-camera mobile phone. *ACM Transactions on Graphics*, 37(4):64, 2018. 1

[45] Wenguan Wang, Jianbing Shen, Fang Guo, Ming-Ming Cheng, and Ali Borji. Revisiting video saliency: A large-scale benchmark and a new model. In *CVPR*, 2018. 5

[46] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *NIPS*, 2015. 5

[47] Yinda Zhang, Neal Wadhwa, Sergio Orts-Escolano, Christian Häne, Sean Fanello, and Rahul Garg. Du2net: Learning depth estimation from dual-cameras and dual-pixels. *ECCV*, 2020. 1

[48] Wenda Zhao, Fan Zhao, Dong Wang, and Huchuan Lu. Defocus blur detection via multi-stream bottom-top-bottom fully convolutional network. In *CVPR*, 2018. 2