Defocus Deblurring Using Dual-Pixel Data

Abdullah Abuolaim[1] and Michael S. Brown[1,2]
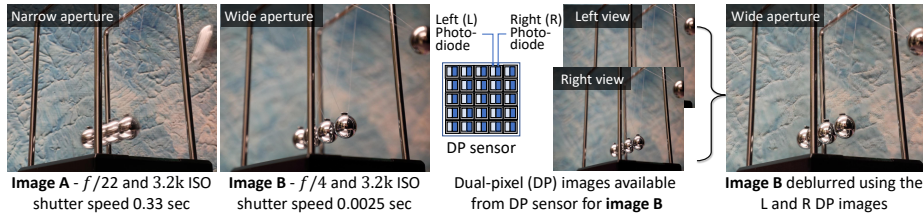
[1] York University, Toronto, Canada
[2] Samsung AI Center, Toronto, Canada
{abuolaim,mbrown}@eecs.yorku.ca

Abstract. Defocus blur arises in images that are captured with a shallow depth of field due to the use of a wide aperture. Correcting defocus blur is challenging because the blur is spatially varying and difficult to estimate. We propose an effective defocus deblurring method that exploits data available on dual-pixel (DP) sensors found on most modern cameras. DP sensors are used to assist a camera's auto-focus by capturing two sub-aperture views of the scene in a single image shot. The two sub-aperture images are used to calculate the appropriate lens position to focus on a particular scene region and are discarded afterwards. We introduce a deep neural network (DNN) architecture that uses these discarded sub-aperture images to reduce defocus blur. A key contribution of our effort is a carefully captured dataset of 500 scenes (2000 images) where each scene has: (i) an image with defocus blur captured at a large aperture; (ii) the two associated DP sub-aperture views; and (iii) the corresponding all-in-focus image captured with a small aperture. Our proposed DNN produces results that are significantly better than conventional single image methods in terms of both quantitative and perceptual metrics – all from data that is already available on the camera but ignored.

Keywords: Defocus blur, extended depth of field, dual-pixel sensors

1 Introduction

This paper addresses the problem of defocus blur. To understand why defocus blur is difficult to avoid, it is important to understand the mechanism governing image exposure. An image's exposure to light is controlled by adjusting two parameters: shutter speed and aperture size. The shutter speed controls the duration of light falling on the sensor, while the aperture controls the amount of light passing through the lens. The reciprocity between these two parameters allows the same exposure to occur by fixing one parameter and adjusting the other. For example, when a camera is placed in *aperture-priority* mode, the aperture remains fixed while the shutter speed is adjusted to control how long light is allowed to pass through the lens. The drawback is that a slow shutter speed can result in motion blur if the camera and/or an object in the scene moves while the shutter is open, as shown in Fig. 1. Conversely, in *shutter-priority* mode, the shutter speed remains fixed while the aperture adjusts its size. The

| Image A - $f/22$ and 3.2k ISO shutter speed 0.33 sec | Image B - $f/4$ and 3.2k ISO shutter speed 0.0025 sec | Dual-pixel (DP) images available from DP sensor for **image B** | **Image B** deblurred using the L and R DP images |

Fig. 1. Images A and B are of the same scene and same approximate exposure. Image A is captured with a narrow aperture (f/22) and slow shutter speed. Image A has a wide depth of field (DoF) and little defocus blur, but exhibits motion blur from the moving object due to the long shutter speed. Image B is captured with a wide aperture (f/4) and a fast shutter speed. Image B exhibits defocus blur due to the shallow DoF, but has no motion blur. Our proposed DNN uses the two sub-aperture views from the dual-pixel sensor of image B to deblur image B, resulting in a much sharper image.

drawback of a variable aperture is that a wide aperture results in a shallow depth of field (DoF), causing defocus blur to occur in scene regions outside the DoF, as shown in Fig. 1. There are many computer vision applications that require a wide aperture but still want an all-in-focus image. An excellent example is cameras on self-driving cars, or cameras on cars that map environments, where the camera must use a fixed shutter speed and the only way to get sufficient light is a wide aperture at the cost of defocus blur.

Our aim is to reduce the unwanted defocus blur. The novelty of our approach lies in the use of data available from dual-pixel (DP) sensors used by modern cameras. DP sensors are designed with two photodiodes at each pixel location on the sensor. The DP design provides the functionality of a simple two-sample light-field camera and was developed to improve how cameras perform autofocus. Specifically, the two-sample light-field provides two sub-aperture views of the scene, denoted in this paper as *left* and *right* views. The light rays coming from scene points that are within the camera's DoF (i.e., points that are in focus) will have no difference in phase between the left and right views. However, light rays coming from scene points outside the camera's DoF (i.e., points that are out of focus) will exhibit a detectable disparity in the left/right views that is directly correlated to the amount of defocus blur. We refere to it as *defocus disparity*. Cameras use this phase shift information to determine how to move the lens to focus on a particular location in the scene. After autofocus calculations are performed, the DP information is discarded by the camera's hardware.

Contribution. We propose a deep neural network (DNN) to perform defocus deblurring that uses the DP images from the sensor available at capture time. In order to train the proposed DNN, a new dataset of 500 carefully captured images exhibiting defocus blur and their corresponding all-in-focus image is collected. This dataset consists of 2000 images – 500 DoF blurred images with their 1000 DP sub-aperture views and 500 corresponding all-in-focus images – all at full-frame resolution (i.e., 6720×4480 pixels). Using this training data, we pro-

pose a DNN architecture that is trained in an end-to-end manner to directly estimate a sharp image from the left/right DP views of the defocused input image. Our approach is evaluated against conventional methods that use only a single input image and show that our approach outperforms the existing state-of-the-art approaches in both signal processing and perceptual metrics. Most importantly, the proposed method works by using the DP sensor images that are a free by-product of modern image capture.

2   Related work

Related work is discussed regarding (1) defocus blur, (2) datasets, and (3) applications exploiting DP sensors.

Defocus deblurring. Related methods in the literature can be categorized into: (1) defocus detection methods [8, 27, 31, 34, 37, 38] or (2) defocus map estimation and deblurring methods [4, 15, 18, 22, 28]. While defocus detection is relevant to our problem, we focus on the latter category as these methods share the goal of ultimately producing a sharp deblurred result.

A common strategy for defocus deblurring is to first compute a defocus map and use that information to guide the deblurring. Defocus map estimation methods [4, 15, 18, 22, 28] estimate the amount of defocus blur per pixel for an image with defocus blur. Representative works include Karaali et al. [15], which uses image gradients to calculate the blur amount difference between the original image edges and their re-blurred ones. Park et al. [22] introduced a method based on hand-crafted and deep features that were extracted from a pre-trained blur classification network. The combined feature vector was fed to a regression network to estimate the blur amount on edges and then later deblur the image. Shi et al. [28] proposed an effective blur feature using a sparse representation and image decomposition to detect just noticeable blur. Methods that directly deblur the image include Andrès et al.'s [4] approach, which uses regression trees to deblur the image. Recent work by Lee et al. [18] introduced a DNN architecture to estimate an image defocus map using a domain adaptation approach. This approach also introduced the first large-scale dataset for DNN-based training. Our work is inspired by Lee et al.'s [18] success in applying DNNs for the DoF deblurring task. Our distinction from the prior work is the use of the DP sensor information available at capture time.

Defocus blur datasets. There are several datasets available for defocus deblurring. The CUHK [27] and DUT [37] datasets have been used for blur detection and provide real images with their corresponding binary masks of blur/sharp regions. The SYNDOF [18] dataset provided data for defocus map estimation, in which their defocus blur is synthesized based on a given depth map of pinhole image datasets. The datasets of [18, 27, 37] do not provide the corresponding ground truth all-in-focus image. The RTF [4] dataset provided light-field images captured by a Lytro camera for the task of defocus deblurring. In their data, each blurred image has a corresponding all-in-focus image. However, the RTF dataset is small, with only 22 image pairs. While there are other similar and much larger
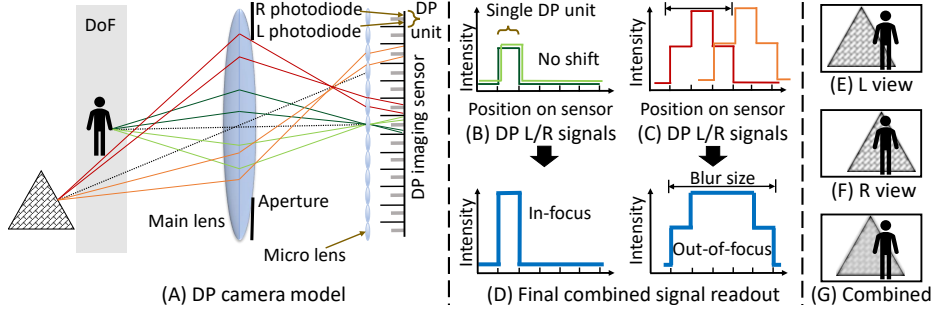
Fig. 2. Image formation diagram for a DP sensor. (A) Shows a thin-lens camera and a DP sensor. The light rays from different halves of the main lens fall on different left and right photodiodes. (B) Scene points that are within the DoF (highlighted in gray) have no phase shift between their L/R views. Scene points outside DoF have a phase shift as shown in (C). The L/R signals are aggregated and the corresponding combined signal is shown in (D). The blur size of the L signal is smaller than the combined one in the out-of-focus case. The defocus disparity is noticeable between the captured L/R images (see (E) and (F)). The final combined image in (G) has more blur. Our DNN leverages this additional information available in the L/R views for image defocus deblurring.

light-field datasets [11,29], these datasets were introduced for different tasks (i.e., depth from focus and synthesizing a 4D RGBD light field), which are different from the task of this paper. In general, the images captured by Lytro cameras are not representative of DSLR and smartphone cameras, because they apply synthetic defocus blur, and have a relatively small spatial resolution [3].

As our approach is to utilize the DP data for defocus deblurring, we found it necessary to capture a new dataset. Our DP defocus blur dataset provides 500 pairs of images of unrepeated scenes; each pair has a defocus blurred image with its corresponding sharp image. The two DP views of the blurred image are also provided, resulting in a total of 2000 images. Details of our dataset capture are provided in Sec. 4. Similar to the patch-wise training approach followed in [18,22], we extract a large number of image patches from our dataset to train our DNN.

DP sensor applications. The DP sensor design was developed by Canon for the purpose of optimizing camera autofocus. DP sensors perform what is termed *phase difference autofocus* (PDAF) [1, 2, 14], in which the phase difference between the left and right sub-aperture views of the primary lens is calculated to measure the blur amount. Using this phase information, the camera's lens is adjusted such that the blur is minimized. While intended for autofocus, the DP images have been found useful for other tasks, such as depth map estimation [6, 24], reflection removal [25], and synthetic DoF [32]. Our work is inspired by these prior methods and examines the use of DP data for the task of defocus blur removal.
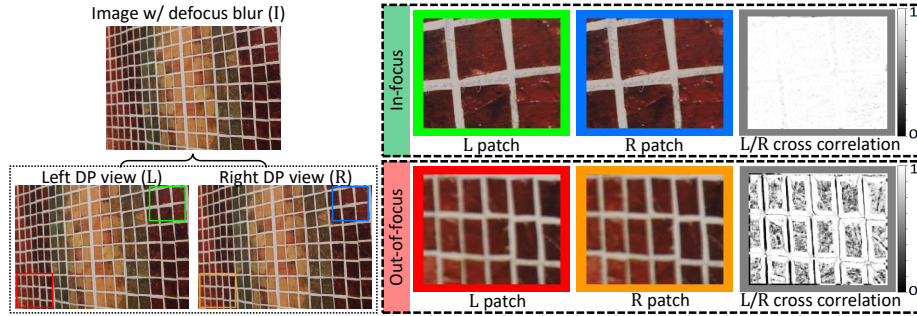
Fig. 3. An input image I is shown with a spatially varying defocus blur. The two dual-pixel (DP) images (L and R) corresponding to I are captured at imaging time. In-focus and out-of-focus patches in the L and R DP image patches exhibit different amounts of pixel disparity as shown by the cross-correlation of the two patches. This information helps the DNN to learn the extent of blur in different regions of the image.

3 DP image formation

We begin with a brief overview of the DP image formation. As previously mentioned, the DP sensor was designed to improve camera auto-focus technology. Fig. 2 shows an illustrative example of how DP imaging works and how the left/right images are formed. A DP sensor provides a pair of photodiodes for each pixel with a microlens placed at the pixel site, as shown in Fig. 2-A. This DP unit arrangement allows each pair of photodiodes (i.e., dual-pixel) to record the light rays independently. Depending on the sensor's orientation, this arrangement can be shown as left/right or top/down pair; in this paper, we refer to them as the left/right pair – or L and R. The difference between the two views is related to the defocus amount at that scene point, where out-of-focus scene points will have a difference in phase and be blurred in opposite directions using a point spread function (PSF) and its flipped one [24]. This difference yields noticeable defocus disparity that is correlated to the amount of defocus blur.

The phase-shift process is illustrated in Fig. 2. The person shown in Fig. 2-A is within the camera's DoF, as highlighted in gray, whereas the textured pyramid is outside the DoF. The light rays from the in-focus object converge at a single DP unit on the imaging sensor, resulting in an in-focus pixel and no disparity between their DP L/R views (Fig. 2-B). The light rays coming from the out-of-focus regions spread across multiple DP units and therefore produce a difference between their DP L/R views, as shown in Fig. 2-C. Intuitively, this information can be exploited by a DNN to learn where regions of the image exhibit blur and the extent of this blur. The final output image is a combination of the L/R views, as shown in Fig. 2-G.

By examining real examples shown in Fig. 3 it becomes apparent how a DNN can leverage these two sub-aperture views as input to deblur the image. In particular, patches containing regions that are out-of-focus will exhibit a notable defocus disparity in the two views that is directly correlated to the amount of

Focal length: 93mm	Focus distance: 1.46m−1.59m	ISO: 100
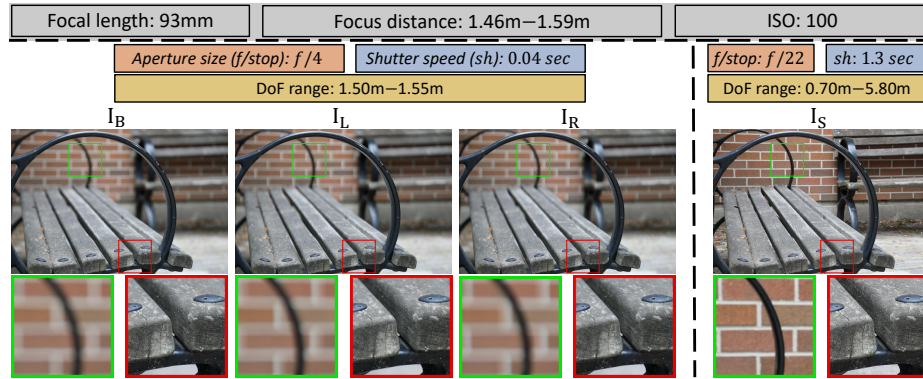
Fig. 4. An example of an image pair with the camera settings used for capturing. I_L and I_R represent the Left and Right DP views extracted from I_B. The focal length, ISO, and focus distance are fixed between the two captures of I_B and I_S. The aperture size is different, and hence the shutter speed and DoF are accordingly different too. In-focus and out-of-focus zoomed-in patches are extracted from each image and shown in green and red boxes, respectively.

defocus blur. By training a DNN with sufficient examples of the L/R views and the corresponding all-in-focus image, the DNN can learn how to detect and correct blurred regions. Animated examples of the difference between the DP views are provided in the supplemental materials.

4   Dataset collection

Our first task is to collect a dataset with the necessary DP information for training our DNN. While most consumer cameras employ PDAF sensors, we are aware of only two camera manufacturers that provide DP data – Google and Canon. Specifically, Google's research team has released an application to read DP data [9] from the Google Pixel 3 and 4 smartphones. However, smartphone cameras are currently not suitable for our problem for two reasons. First, smartphone cameras use fixed apertures that cannot be adjusted for data collection. Second, smartphone cameras have narrow aperture and exhibit large DoF; in fact, most cameras go to great lengths to simulate shallow DoF by purposely introducing defocus blur [32]. As a result, our dataset is captured using a Canon EOS 5D Mark IV DSLR camera, which provides the ability to save and extract full-frame DP images.

Using the Canon camera, we capture a pair of images of the same static scene at two aperture sizes – $f/4$ and $f/22$ – which are the maximum (widest) and minimum (narrowest) apertures possible for our lens configuration. The lens position and focal length remain fixed during image capture. Scenes are captured in aperture-priority mode, in which the exposure compensation between the

image pairs is done automatically by adjusting the shutter speed. The image captured at $f/4$ has the smallest DoF and results in the blurred input image I_B. The image captured at $f/22$ has the largest DoF and serves as the all-in-focus target image denoted as I_S (sharp image). Focus distance and focal length differ across captured pairs in order to capture a diverse range of defocus blur types. Our captured images offer the following benefits over prior datasets:

High-quality images. Our captured images are low-noise images (i.e., low ISO equates to low-noise [23]) and of full resolution of 6720×4480. All images, including the left/right DP views, are processed to an sRGB and encoded with a lossless 16-bit depth per RGB channel.

Real and diverse defocus blur. Unlike other existing datasets, our dataset provides real defocus blur and in-focus pairs indicative of real camera optics.

Varying scene contents. To provide a wide range of object categories, we collect 500 pairs of unique indoor/outdoor scenes with a large variety of scene contents. Our dataset is also free of faces to avoid privacy issues.

The $f/4$ (blurry) and $f/22$ (sharp) image pairs are carefully imaged static scenes with the camera fixed on a tripod. To further avoid camera shake, the camera was controlled remotely to allow hands-free operation. Fig. 4 shows an example of an image pair from our dataset. The left and right DP views of I_B are provided by the camera and denoted as I_L and I_R respectively. The ISO setting is fixed for each image pair. Fig. 4 shows the DP L/R views for only image I_B, because DP L/R views of I_S are visually identical due to the fact I_S is our all-in-focus ground truth.

5   Dual-pixel defocus deblurring DNN (DPDNet)

Using our captured dataset, we trained a symmetric encoder-decoder CNN architecture with skip connections between the corresponding feature maps [20, 26]. Skip connections are widely used in encoder-decoder CNNs to combine various levels of feature maps. These have been found useful for gradient propagation and convergence acceleration that allow training of deeper networks [13, 30].

We adapt a U-Net-like architecture [26] with the following modifications: an input layer to take a 6-channel input cube (two DP views; each is a 3-channel sRGB image) and an output layer to generate a 3-channel output sRGB image; skip connections of the convolutional feature maps are passed to their mirrored convolutional layers without cropping in order to pass on more feature map detail; and the loss function is changed to be mean squared error (MSE).

The overall DNN architecture of our proposed DP deblurring method is shown in Fig. 5. Our method reads the two DP images, I_L and I_R, as a 6-channel cube, and processes them through the encoder, bottleneck, and decoder stages to get the final sharp image I_S^*. There are four blocks in the encoder stage (E-Block 1–4) and in each block, two 3×3 convolutional operations are performed, each followed by a ReLU activation. Then a 2×2 max pooling is performed for downsampling. Although max pooling operations reduce the size
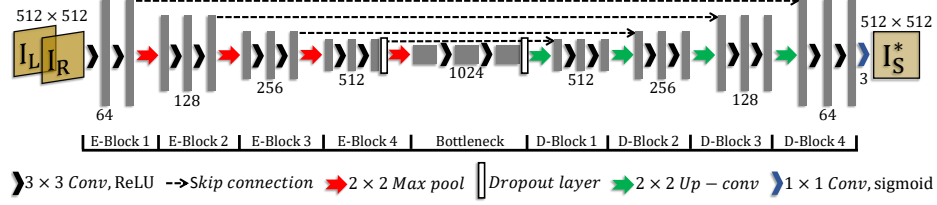
Fig. 5. Our proposed DP deblurring architecture (DPDNet). Our method utilizes the DP images, I_L and I_R, for predicting the sharp image I_S^* through three stages: encoder (E-Blocks), bottleneck, and decoder (D-Blocks). The size of the input and output layers is shown above the images. The number of output filters is shown under the convolution operations for each block.

of feature maps between E-Blocks, this is required to extend the receptive field size in order to handle large defocus blur. To reduce the chances of overfitting, two dropout layers are added, one before the max pooling operation in the fourth E-Block, and one dropout layer at the end of the network bottleneck, as shown in Fig. 5. In the decoder stage, we also have four blocks (D-Block 1–4). For each D-Block, a 2×2 upsampling of the input feature map followed by a 2×2 convolution $(Up - conv)$ is carried out instead of direct deconvolution in order to avoid checkerboard artifacts [21]. The corresponding feature map from the encoder stage is concatenated. Next, two 3×3 convolutions are performed, each followed by a ReLU activation. Afterwards, a 1×1 convolution followed by sigmoid activation is applied to output the final sharp image I_S^*. The number of output filters is shown under each convolution layer for each block in Fig. 5. The stride for all operations is 1 except for the max pooling operation, which has a stride of 2. The final sharp image I_S^* is, thus, predicted as follows:

$$I_S^* = \text{DPDNet}(I_L, I_R; \theta_{\text{DPDNet}}), \tag{1}$$

where DPDNet is our proposed architecture, and θ_{DPDNet} is the set of weights and parameters.

Training procedure. The size of input and output layers is set to $512 \times 512 \times 6$ and $512 \times 512 \times 3$, respectively. This is because we train not on the full-size images but on the extracted image patches. We adopt the weight initialization strategy proposed by He [12] and use the Adam optimizer [16] to train the model. The initial learning rate is set to 2×10^{-5}, which is decreased by half every 60 epochs. We train our model with mini-batches of size 5 using MSE loss between the output and the ground truth as follows:

$$\mathcal{L} = \frac{1}{n} \sum_n (I_S - I_S^*)^2, \tag{2}$$

where n is the size of the image patch in pixels. During the training phase, we set the dropout rate to 0.4. All the models described in the subsequent sections are implemented using Python with the Keras framework on top of TensorFlow

Method	Indoor				Outdoor				Combined			
	PSNR ↑	SSIM ↑	MAE ↓	LPIPS ↓	PSNR ↑	SSIM ↑	MAE ↓	LPIPS ↓	PSNR ↑	SSIM ↑	MAE ↓	LPIPS ↓
EBDB [15]	25.77	0.772	0.040	0.297	21.25	0.599	0.058	0.373	23.45	0.683	0.049	0.336
DMENet [18]	25.50	0.788	0.038	0.298	21.43	0.644	0.063	0.397	23.41	0.714	0.051	0.349
JNB [28]	26.73	0.828	0.031	0.273	21.10	0.608	0.064	0.355	23.84	0.715	0.048	0.315
Our DPDNet-Single	26.54	0.816	0.031	0.239	22.25	0.682	0.056	0.313	24.34	0.747	0.044	0.277
Our DPDNet	27.48	0.849	0.029	0.189	22.90	0.726	0.052	0.255	25.13	0.786	0.041	0.223

Table 1. The quantitative results for different defocus deblurring methods. The testing on the dataset is divided into three scene categories: indoor, outdoor, and combined. The top result numbers are highlighted in green and the second top in blue. DPDNet-Single is our DPDNet variation that is trained with only a single blurred input. Our DPDNet that uses the two L/R DP views achieved the best results on all scene categories for all metrics. Note: the testing set consists of 37 indoor and 39 outdoor scenes.

and trained with a NVIDIA TITAN X GPU. We set the maximum number of training epochs to 200.

6   Experimental results

We first describe our data preparation procedure and then evaluation metrics used. This is followed by quantitative and qualitative results to evaluate our proposed method with existing deblurring methods. We also discuss the time analysis and test the robustness of our DP method against different aperture settings.

Data preparation. Our dataset has an equal number of indoor and outdoor scenes. We divide the data into 70% training, 15% validation, and 15% testing sets. Each set has a balanced number of indoor/outdoor scenes. To prepare the data for training, we first downscale our images to be 1680×1120 in size. Next, image patches are extracted by sliding a window of size 512×512 with 60% overlap. We empirically found this image size and patch size to work well. An ablation study of different architecture settings is provided in the supplemental materials. We compute the sharpness energy (i.e., by applying Sobel filter) of the in-focus image patches and sort them. We discard 30% of the patches that have the lowest sharpness energy. Such patches represent homogeneous regions, cause an ambiguity associated to the amount of blur, and adversely affect the DNNs training, as found in [22].

Evaluation metrics. Results are reported on traditional signal processing metrics – namely, PSNR, SSIM [33], and MAE. We also incorporate the recent learned perceptual image patch similarity (LPIPS) proposed by [35]. The LPIPS metric is correlated with human perceptual similarity judgments as a perceptual metric for low-level vision tasks, such as enhancement and image deblurring.

Quantitative results. We compare our DPDNet with the following three methods: the edge-based defocus blur (EBDB) [15], the defocus map estimation network (DMENet) [18], and the just noticeable blur (JNB) [28] estimation. These methods accept only a single image as input – namely, I_B – and estimate the

Fig. 6. Qualitative comparisons of different deblurring methods. The first row is the input image that has a spatially varying blur, and the last row is the corresponding ground truth sharp image. The rows in between are the results of different methods. We also present zoomed-in cropped patches in green and red boxes. Our DPDNet method significantly outperforms other methods in terms of deblurring quality.
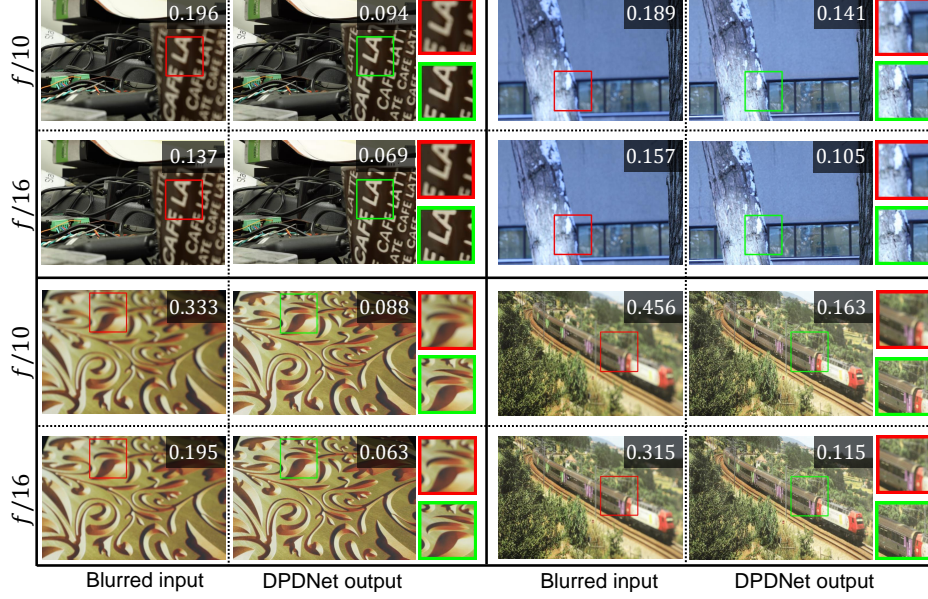
Fig. 7. Examining DPDNet's robustness to different aperture settings. Four scenes are presented; each has two different apertures. In each scene, the left-hand image is the blurred one, I_B, and the right-hand image is the deblurred one, I_S^*, computed by our DPDNet. The number shown on each image is the LPIPS measure compared with the ground truth I_S. Zoomed-in cropped patches are also provided. Even though our training data was on blurry examples with an $f/4$ aperture, our DPDNet method is able to generalize well to different aperture settings.

defocus map in order to use it to guide the deblurring process. The EBDB [15] and JNB [28] are not learning-based methods. We test them directly on our dataset using I_B as input. The EBDB uses a combination of non-blind deblurring methods proposed in [17, 19], and for a fair comparison, we contacted the authors for their deblurring settings and implementation. The JNB method uses the non-blind defocus deblurring method from [5].

For the deep-learning-based method (i.e., DMENet [18]), the method requires the ground truth defocus map for training. In our dataset, we do not have this ground truth defocus map and provide only the sharp image, since our approach in this work is to solve directly for defocus deblurring. Therefore, we tested the DMENet on our dataset using I_B as input without retraining. For deblurring, DMENet adopts a non-blind deconvolution algorithm proposed by [17]. Our results are compared against code provided by the authors. Unfortunately, the methods in [4, 22] do not have the deblurring code available for comparison.

To show the advantage of utilizing DP data for defocus deblurring, we introduce a variation of our DPDNet that accepts only a single input (i.e., I_B) and uses exactly the same architecture settings along with the same training

procedure as shown in Fig 5. We refer to this variation as DPDNet-Single in Table 1. Our proposed architecture is fully convolutional, which enables testing any image size during the testing phase. Therefore, all the subsequent results are reported on the testing set using the full image for all methods. Table 1 reports our findings by testing on three scene categories: indoor, outdoor, and combined. Top result numbers are highlighted in green and the second top ones in blue. Our DPDNet method has a significantly better deblurring ability based on all metrics for all testing categories. Furthermore, DP data is the key that made our DPDNet method outperforms others, especially the single image input one (i.e., DPDNet-Single), in which it has exactly the same architecture but does not utilize DP views. Interestingly, all methods have better deblurring results for indoor scenes, due to the fact that outdoor scenes tend to have larger depth variations, and thereby more defocus blur.

Qualitative results. In Fig. 6, we present the qualitative results of different defocus deblurring methods. The first row shows the input image with a spatially varying defocus blur; the last row shows the corresponding ground truth sharp image. The rows in between present different methods, including ours. This figure also shows two zoomed-in cropped patches in green and red to further illustrate the difference visually. From the visual comparison with other methods, our DPDNet has the best deblurring ability and is quite similar to the ground truth. EBDB [15], DMENet [18], and JNB [28] are not able to handle spatially varying blur with almost unnoticeable difference with the input image. EBDB [15] tends to introduce some artifacts in some cases. Our single image method (i.e., DPDNet-Single) has better deblurring ability compared to other traditional deblurring methods, but it is not at the level of our method that utilizes DP views for deblurring. Our DPDNet, as shown visually, is effective in handling spatially varying blur. For example, in the second row, the image has a part that is in focus and another is not; our DPDNet method is able to determine the deblurring amount required for each pixel, in which the in-focus part is left untouched. Further qualitative results are provided in our supplemental materials, including results on DP data obtained from a smartphone camera.

Time analysis. We examine evaluating different defocus deblurring methods based on the time required to process a testing image of size 1680×1120 pixels. Our DPDNet directly computes the sharp image in a single pass, whereas other methods [15,18,28] use two passes: (1) defocus map estimation and (2) non-blind deblurring based on the estimated defocus map.

Non-learning-based methods (i.e., EBDB [15] and JNB [28]) do not utilize the GPU and use only the CPU. For the deep-learning method (i.e., DMENet [18]), it utilizes the GPU for the first pass; however, the deblurring routine is applied on a CPU. This time evaluation is performed using Intel Core i7-6700 CPU and NVIDIA TITAN X GPU. Our DPDNet operates in a single pass and can process the testing image of size 1680×1120 pixels about 1.2×10^3 times faster compared to the second-best method (i.e., DMENet), as shown in Table 2.

Robustness to different aperture settings. In our dataset, the image pairs are captured using aperture settings corresponding to f-stops $f/22$ and $f/4$.

Method	Time (Sec) $\downarrow$		
	Defocus map estimation	Defocus deblurring	Total
EBDB [15]	57.2	872.5	929.7
DMENet [18]	1.3	612.4	613.7
JNB [28]	605.4	237.7	843.1
Our DPDNet	**0**	**0.5**	**0.5**

Table 2. Time analysis of different defocus deblurring methods. The last column is the total time required to process a testing image of size 1680×1120 pixels. Our DPDNet is about 1.2×10^3 times faster compared to the second-best method (i.e., DMENet).

Recall that $f/4$ results in the greatest DoF and thus most defocus blur. Our DPDNet is trained on diverse images with many different depth values; thus, our training data spans the worst-case blur that would be observed with any aperture settings. To test the ability of our DPDNet in generalizing for scenes with different aperture settings, we capture image pairs with aperture settings $f/10$ and $f/16$ for the blurred image and again $f/22$ for the corresponding ground truth image. Our DPDNet is applied to these less blurred images. Fig. 7 shows the results for four scenes, where each scene's image has its LPIPS measure compared with the ground truth. For better visual comparison, Fig. 7 provides zoomed-in patches that are cropped from the blurred input (red box) and the deblurred one (green box). These results show that our DPDNet is able to deblur scenes with different aperture settings that have not been used during training.
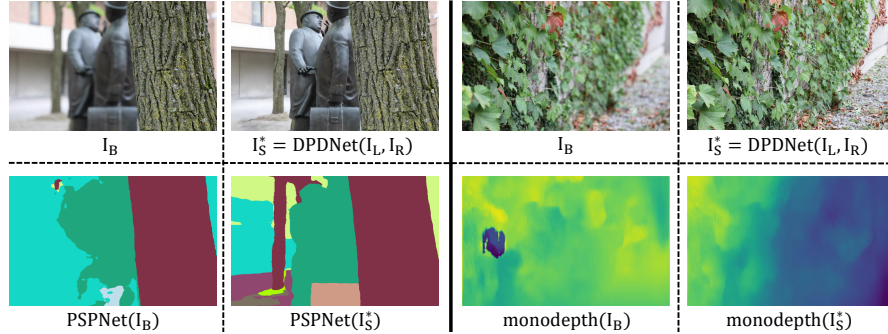
7   Applications

Image blur can have a negative impact on some computer vision tasks, as found in [10]. Here we investigate defocus blur effect on two common computer vision tasks – namely, image segmentation and monocular depth estimation.

Image segmentation. The first two columns in Fig. 8 demonstrate the negative effect of defocus blur on the task of image segmentation. We use the PSPNet segementation model from [36], and test two images: one is the blurred input image I_B and another is the deblurred one I_S^* using our DPDNet deblurring model. The segmentation results are affected by I_B – only the foreground tree was correctly segmented. PSPNet assigns cyan color to unknown categories, where a large portion of I_B is segmented as unknown. On the other hand, the segmentation results of I_S^* are much better, in which more categories are segmented correctly. With that said, image DoF deblurring using our DP method can be beneficial for the task of image segmentation.

Monocular depth estimation. The monocular depth estimation is the task of estimating scene depth using a single image. In the last two columns of Fig. 8, we show the direct effect of defocus blur on this task. We use the monodepth model from [7] to test the two images I_B and I_S^* in order to examine the change in performance. The result of monodepth is affected by the defocus blur, in which the depth map estimated is completely wrong. Contrarily, the result of

| I_B | $I_S^* = \text{DPDNet}(I_L, I_R)$ | I_B | $I_S^* = \text{DPDNet}(I_L, I_R)$ |
| PSPNet(I_B) | PSPNet(I_S^*) | monodepth(I_B) | monodepth(I_S^*) |

Fig. 8. The effect of defocus blur on some computer vision tasks. The first two columns show the image segmentation results using the PSPNet [36] segmentation model. The segmentation results are affected by the blurred image I_B, where a large portion is segmented as unknown in cyan. The last two columns show the results of the monocular depth estimation using the monodepth model from [7]. The depth estimation is highly affected by the defocus blur and produced wrong results. Deblurring I_B using our DP deblurring method has significantly improved the results for both tasks.

monodepth has been significantly improved after testing with the deblurred input image using our DPDNet deblurring model. Therefore, deblurring images using our DPDNet can be useful for the task of monocular depth map estimation.

8   Conclusion

We have presented a novel approach to reduce the effect of defocus blur present in images captured with a shallow DoF. Our approach leverages the DP data that is available in most modern camera sensors but currently being ignored for other uses. We show that the DP images are highly effective in reducing DoF blur when used in a DNN framework. As part of this effort, we have captured a new image dataset consisting of blurred and sharp image pairs along with their DP images. Experimental results show that leveraging the DP data provides state-of-the-art quantitative results on both signal processing and perceptual metrics. We also demonstrate that our deblurring method can be beneficial for other computer vision tasks. We believe our captured dataset and DP-based method are useful for the research community and will help spur additional ideas about both defocus deblurring and applications that can leverage data from DP sensors. The dataset, code, and trained models are available at https://github.com/Abdullah-Abuolaim/defocus-deblurring-dual-pixel.

References

1. Abuolaim, A., Brown, M.S.: Online lens motion smoothing for video autofocus. In: WACV (2020)
2. Abuolaim, A., Punnappurath, A., Brown, M.S.: Revisiting autofocus for smartphone cameras. In: ECCV (2018)
3. Boominathan, V., Mitra, K., Veeraraghavan, A.: Improving resolution and depth-of-field of light field cameras using a hybrid imaging system. In: ICCP (2014)
4. D'Andrès, L., Salvador, J., Kochale, A., Süsstrunk, S.: Non-parametric blur map regression for depth of field extension. TIP **25**(4), 1660–1673 (2016)
5. Fish, D., Brinicombe, A., Pike, E., Walker, J.: Blind deconvolution by means of the richardson–lucy algorithm. Journal of the Optical Society of America (A) **12**(1), 58–65 (1995)
6. Garg, R., Wadhwa, N., Ansari, S., Barron, J.T.: Learning single camera depth estimation using dual-pixels. In: ICCV (2019)
7. Godard, C., Mac Aodha, O., Brostow, G.J.: Unsupervised monocular depth estimation with left-right consistency. In: CVPR (2017)
8. Golestaneh, S.A., Karam, L.J.: Spatially-varying blur detection based on multiscale fused and sorted transform coefficients of gradient magnitudes. In: CVPR (2017)
9. Google: Google research: Android app to capture dual-pixel data. https://github.com/google-research/google-research/tree/master/dual_pixels (2019), last accessed: March, 2020
10. Guo, Q., Feng, W., Chen, Z., Gao, R., Wan, L., Wang, S.: Effects of blur and deblurring to visual object tracking. arXiv preprint arXiv:1908.07904 (2019)
11. Hazirbas, C., Soyer, S.G., Staab, M.C., Leal-Taixé, L., Cremers, D.: Deep depth from focus. In: ACCV (2018)
12. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: ICCV (2015)
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
14. Jang, J., Yoo, Y., Kim, J., Paik, J.: Sensor-based auto-focusing system using multi-scale feature extraction and phase correlation matching. Sensors **15**(3), 5747–5762 (2015)
15. Karaali, A., Jung, C.R.: Edge-based defocus blur estimation with adaptive scale selection. TIP **27**(3), 1126–1137 (2017)
16. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
17. Krishnan, D., Fergus, R.: Fast image deconvolution using hyper-laplacian priors. In: NeurIPS (2009)
18. Lee, J., Lee, S., Cho, S., Lee, S.: Deep defocus map estimation using domain adaptation. In: CVPR (2019)
19. Levin, A., Fergus, R., Durand, F., Freeman, W.T.: Image and depth from a conventional camera with a coded aperture. ACM Transactions on sGraphics **26**(3), 70 (2007)
20. Mao, X., Shen, C., Yang, Y.B.: Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In: NeurIPS (2016)
21. Odena, A., Dumoulin, V., Olah, C.: Deconvolution and checkerboard artifacts. Distill **1**(10), e3 (2016)
22. Park, J., Tai, Y.W., Cho, D., So Kweon, I.: A unified approach of multi-scale deep and hand-crafted features for defocus estimation. In: CVPR (2017)

23. Plotz, T., Roth, S.: Benchmarking denoising algorithms with real photographs. In: CVPR (2017)
24. Punnappurath, A., Abuolaim, A., Afifi, M., Brown, M.S.: Modeling defocus-disparity in dual-pixel sensors. In: ICCP (2020)
25. Punnappurath, A., Brown, M.S.: Reflection removal using a dual-pixel sensor. In: CVPR (2019)
26. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: MICCAI (2015)
27. Shi, J., Xu, L., Jia, J.: Discriminative blur detection features. In: CVPR (2014)
28. Shi, J., Xu, L., Jia, J.: Just noticeable defocus blur detection and estimation. In: CVPR (2015)
29. Srinivasan, P.P., Wang, T., Sreelal, A., Ramamoorthi, R., Ng, R.: Learning to synthesize a 4D RGBD light field from a single image. In: ICCV (2017)
30. Srivastava, R.K., Greff, K., Schmidhuber, J.: Training very deep networks. In: NeurIPS (2015)
31. Tang, C., Zhu, X., Liu, X., Wang, L., Zomaya, A.: Defusionnet: Defocus blur detection via recurrently fusing and refining multi-scale deep features. In: CVPR (2019)
32. Wadhwa, N., Garg, R., Jacobs, D.E., Feldman, B.E., Kanazawa, N., Carroll, R., Movshovitz-Attias, Y., Barron, J.T., Pritch, Y., Levoy, M.: Synthetic depth-of-field with a single-camera mobile phone. ACM Transactions on Graphics **37**(4), 64 (2018)
33. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P., et al.: Image quality assessment: from error visibility to structural similarity. TIP **13**(4), 600–612 (2004)
34. Yi, X., Eramian, M.: Lbp-based segmentation of defocus blur. TIP **25**(4), 1626–1638 (2016)
35. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR (2018)
36. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: CVPR (2017)
37. Zhao, W., Zhao, F., Wang, D., Lu, H.: Defocus blur detection via multi-stream bottom-top-bottom fully convolutional network. In: CVPR (2018)
38. Zhao, W., Zheng, B., Lin, Q., Lu, H.: Enhancing diversity of defocus blur detectors via cross-ensemble network. In: CVPR (2019)