Learning to Synthesize Photorealistic Dual-pixel Images from RGBD frames

Feiran Li, Heng Guo, *Member, IEEE*, Hiroaki Santo, *Member, IEEE*,
Fumio Okura, *Member, IEEE*, and Yasuyuki Matsushita, *Senior Member, IEEE*

Abstract—As a special sensor that implicitly provides ordinal depth information, dual-pixel (DP) appears to be beneficial for various tasks such as defocus deblurring and monocular depth estimation. Recent advances in data-driven dual-pixel (DP) research are bottlenecked by the difficulties in reaching large-scale DP datasets, and a photorealistic image synthesis approach appears to be a credible solution. To benchmark the accuracy of various existing DP image simulators and facilitate data-driven DP image synthesis, this work presents a real-world DP dataset consisting of approximately 5000 high-quality pairs of sharp images, DP defocus blur images, detailed imaging parameters, and accurate depth maps. Based on this large-scale dataset, we also propose a holistic data-driven framework to synthesize photorealistic DP images, where a neural network replaces conventional handcrafted imaging models. Experiments show that our neural DP simulator can generate more photorealistic DP images than existing state-of-the-art methods and effectively benefit data-driven DP-related tasks. Our code and dataset are released at https://github.com/SILI1994/Dual-Pixel-Simulator.

Index Terms—Dual pixel, Image synthesis, Dataset, Data-driven synthesis

1 Introduction

DUAL-pixel (DP) is a hardware architecture widely used in modern cameras and smartphones to improve speed and accuracy of phase detection auto-focus [1]. Unlike conventional image sensors, a DP sensor can record two images by a single shot, providing focus and ordinal depth information. Owing to this capability, there recently emerges a trend to investigate DP images in computer vision (CV) tasks that can benefit from geometric hints, such as single image depth estimation [2], [3], [4] and defocus deblurring [5], [6], [7].

Despite the great potential of DP images in various computer vision tasks, data-driven approaches using DP images have not advanced well due to difficulties in obtaining DP data. To solve this problem, recent works have explored synthesizing DP images from RGBD frames [3], [8], [9]. These existing methods either use simple handcrafted parametric functions to model the shape of DP point spread functions (PSFs), or assume an ideally manufactured sensor and employ ray-tracing techniques to mimic the light ray splitting process of DP. However, these handcrafted PSF models and the ideal manufacturing assumption may deviate significantly from the real-world imaging process, leaving space for further improving the synthesis accuracy. Also, the relative performances of different DP simulators need to be better studied.

To support data-driven DP research, this paper presents a carefully collected real-world DP image dataset, called the *DP5K* dataset. As shown in Fig. 1, this dataset consists of 5130 aligned pairs of sharp images, DP defocus blur images, accurate depth maps, focusing information, and other imaging-related records. Furthermore, owing to the large size of our DP5K dataset, we develop a holistic data-driven framework for flexibly synthesizing photorealistic DP images from off-the-shelf pinhole RGBD frames.

- *Project done in Osaka University. The first two authors contributed equally.*
- *Feiran Li is with Sony Research. Heng Guo, Hiroaki Santo, Fumio Okura, and Yasuyuki Matsushita are with Osaka University.*
- *Email addresses: feiran.li@sony.com, {heng.guo, santo.hiroaki, okura, yasumat}@ist.osaka-u.ac.jp*

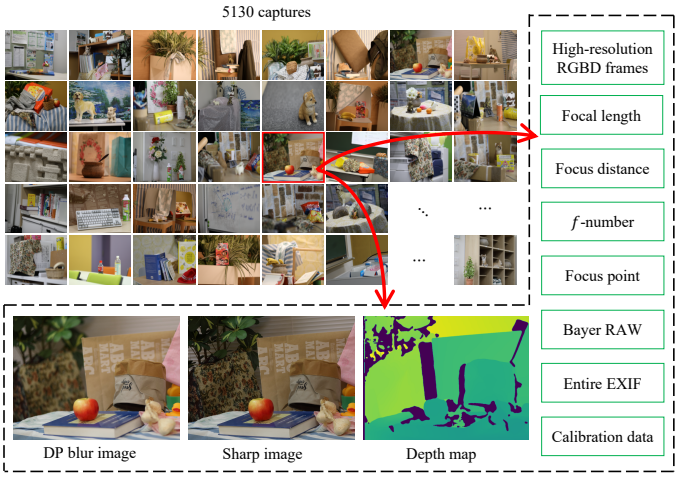


Fig. 1: A conceptual illustration of our DP5K dataset. This datasets contains 5130 pairs of DP defocus blur, sharp, and depth images. The detailed imaging information of each pair is also recorded.

We summarize our contributions as follows:

- We present a real-world DP dataset for advancing the study of DP simulators. As far as we know, this is the first dataset that enables benchmarking and extensive comparison of DP simulators by jointly providing sharp, blur, and depth images and imaging-related parameters. Besides DP simulators, our dataset may also be applied to other studies such as depth from defocus, defocus deblurring, bokeh rendering, and neural image signal processing.
- We propose Neural DP Simulator, a holistic data-driven pipeline for synthesizing photorealistic DP images while allowing for adjustable defocus blur effects, as shown in Fig. 2. Compared to existing DP simulators, our method can present more accurate syntheses and hence better benefits the downstream DP applications.

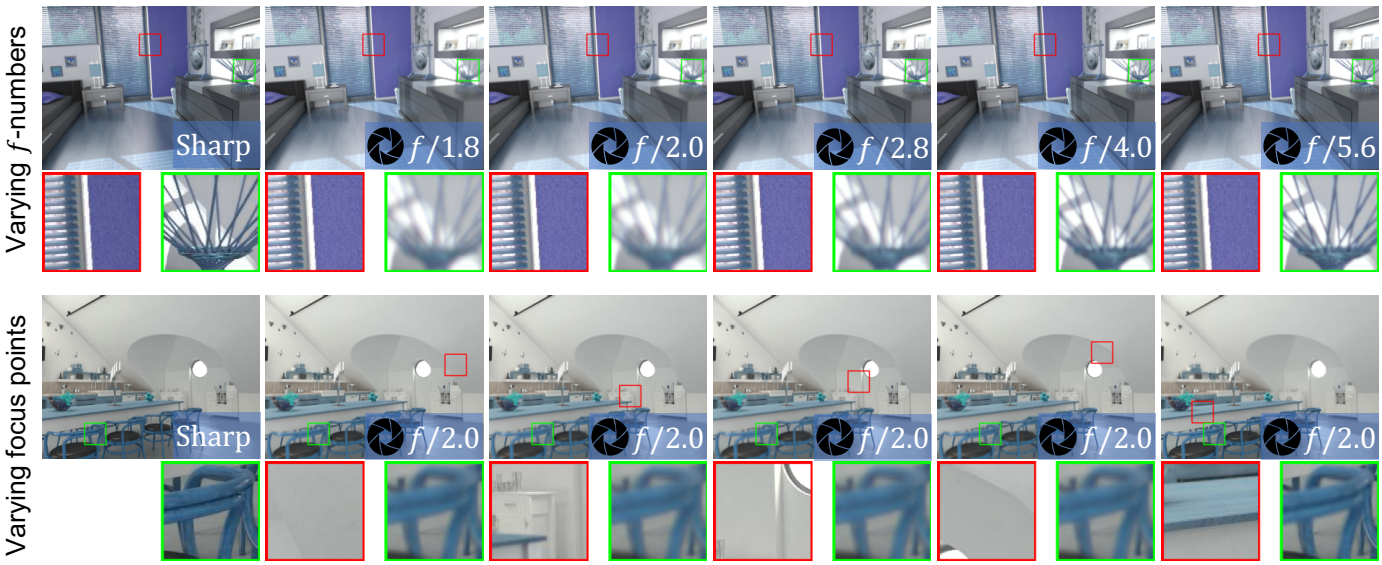


Fig. 2: Example DP images synthesized by our Neural DP Simulator. All images are shown in a combined form (*i.e.*, the sum of the left and right DP views). Red and green squares denote the focused area and blurred area, respectively. Our simulator enables users to flexibly select the f-numbers (top row) and focus distance (bottom row) to simulate different DP defocus blur effects.

	Depth acquisition	Blur & Sharp acquisition	CoC map*	Scenario
Punnappurath *et al*. [3]	Depth-from-defocus	–	✗	Indoor & Outdoor
Garg *et al*. [2]	Multi-view stereo	–	✗	Indoor & Outdoor
Abuolaim *et al*. [5], [6]	–	$f/4, \cdots, f/22$	✗	Indoor & Outdoor
Pan *et al*. [9]	–	$f/4, \cdots, f/22$	✗	Indoor & Outdoor
Kang *et al*. [4]	Structured-light	–	✗	Human face
DP5K (Ours)	Structured-light	$f/1.8, \cdots, f/22$	✓	Indoor

* We define a CoC map akin to a depth map with its entries being the signed CoC radii.

TABLE 1: Summary of existing DP datasets. $f/\cdot$ denotes the f-number.

The remaining sections are structured as follows: In Section , we provide a brief overview of existing works related to DP and learning-based data synthesis techniques. Section 3 presents the details of the DP5K dataset, while Section 4 delves into the specifics of our proposed simulator. In Section 5, we demonstrate the usefulness of our dataset and simulator through experiments. Finally, Section 6 concludes the paper, discussing its limitations and suggesting future directions.

2 Related Works

We here review DP-related applications, existing DP simulators and datasets, and similar learning-based data synthesis techniques.

2.1 DP-related applications

Some works benefit from the ordinal depth information recorded by DP images. For example, it is demonstrated that DP images can effectively boost the accuracy of monocular depth estimation [2], [3], [4], [10]. Punnappurath and Brown [11] employ DP images for reflection removal by treating reflections as backward distanced from the DoF. Wu *et al*. [12] apply DP images to face anti-spoofing to detect planar attacks. There also exist studies that take advantage of the blur-aware property of DP images, leading to plausible defocus deblurring results [5], [6], [7], [8].

2.2 Existing DP simulators and datasets

There are a few methods capable of DP image synthesis, and all of them use RGBD frames as inputs. Specifically, Punnappurath *et al*. [3] emulate the shapes of DP PSFs by a translating disk kernel, which is the sum of several differently centered circles with their radii being the circle-of-confusion (CoC) radii. Abuolaim *et al*. [8] introduce a more precise model of DP PSFs, whose shape and size are jointly decided by a Butterworth filter, a Gaussian filter, and the CoC radius. Pan *et al*. [9] discretize the thin-lens to multiple points to realize DP ray-tracing. Unlike these approaches using handcrafted parametric models to establish the mapping from RGBD frames to DP images, we propose to model the PSFs in a data-driven manner via neural networks, leading to more photorealistic results.

The number of existing DP-related datasets is quite limited. As listed in Table 1, Punnappurath *et al*. [3] present a depth estimation dataset with the ground truths estimated via depth-from-defocus. Garg *et al*. [2] collect a multi-view dataset for depth estimation, whose ground truths are obtained by multi-view stereo. Abuolaim and Brown *et al*. [5] present a defocus deblurring dataset where the sharp and blurry image pairs are captured with different f-numbers and further enlarged it in their following work [6]. Under a similar setup, Pan *et al*. [9] collect a defocus deblurring test set containing both indoor and outdoor scenes. Kang *et al*. [4] present a facial dataset for depth and surface normal estimation, where the ground-truth depth and normal maps are obtained using structured light

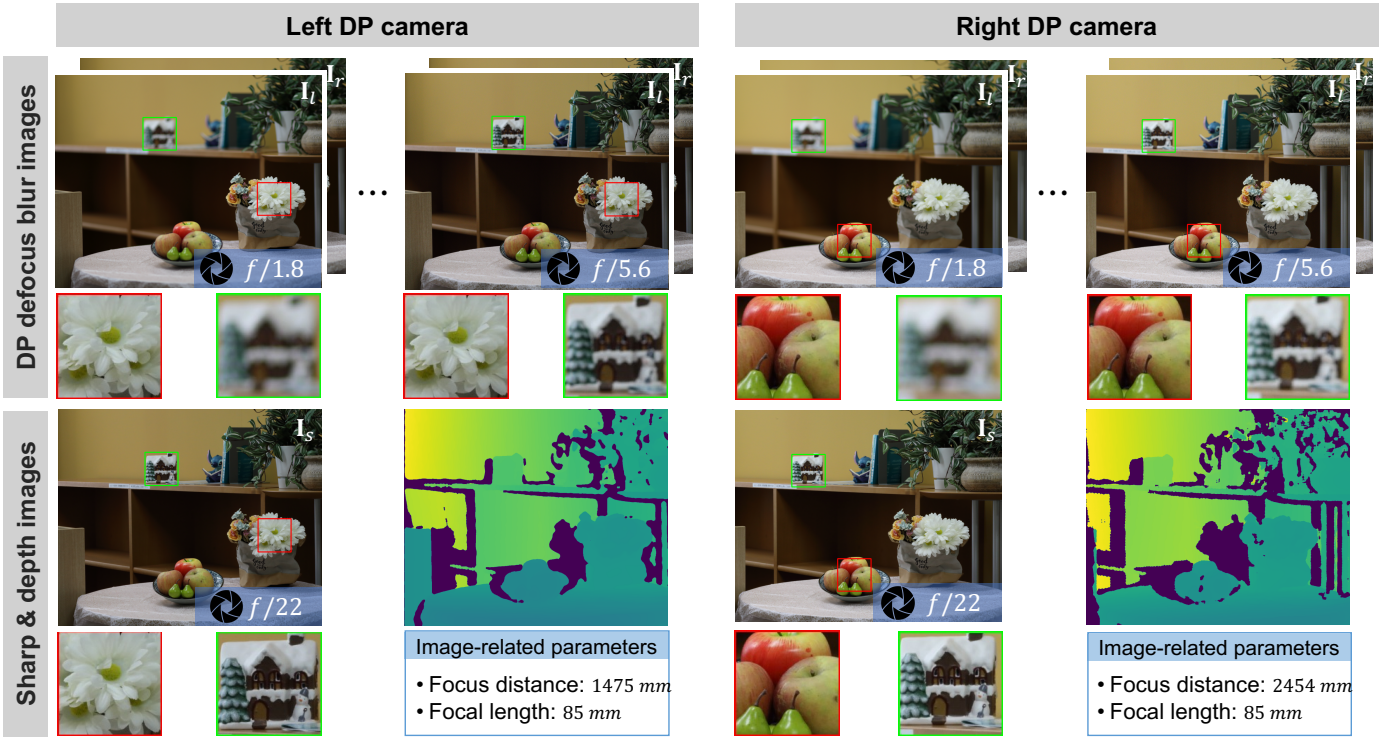


Fig. 3: An example scene of our DP5K dataset. Occluded area between the stereo views is cropped. Animated GIF versions of the DP views are presented in the supplementary material.

and photometric stereo, respectively. Unlike these existing works, our dataset concurrently provides sharp, blur, depth images, and all the necessary information for generating CoC maps, which are necessary for exploring the relations between depth and defocus blur of DP sensors.

2.3 Learning-based data generation

Machine learning is an effective tool for generating data for data-driven approaches. For example, Srinivasan *et al*. [13] develop a convolutional neural network to synthesize 4D RGBD light-field images from single RGB ones. Brooks and Barron [14] present a deep learning approach to synthesize motion blur effects from paired sharp images for the deblurring task. Sandfort *et al*. [15] employ generative adversarial networks to augment the training data for CT image segmentation. In general, these techniques can effectively reduce the amount of real-world data required by data-driven approaches to reach reasonable performances.

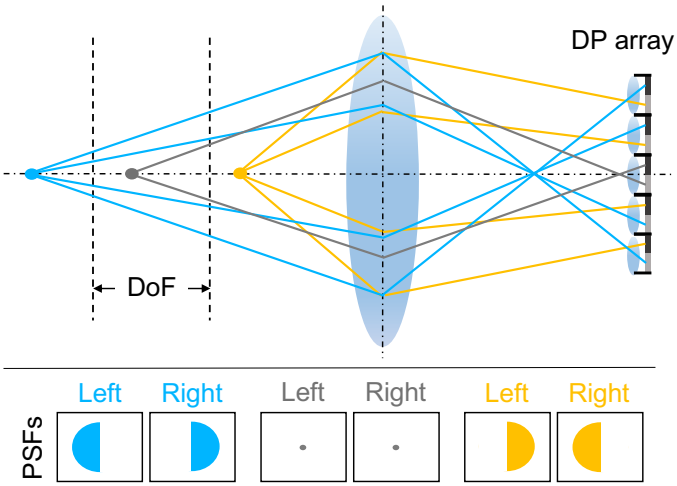


Fig. 4: DP image formation model. A DP camera can be treated as a special light-field one with angular resolution set to 2.

3 DP5K Dataset

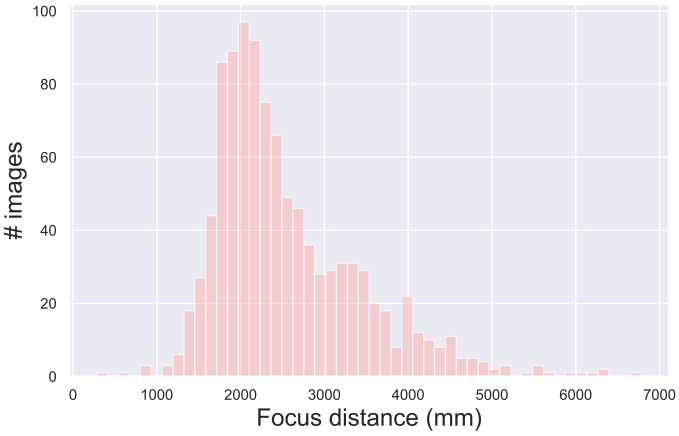
We here introduce the details of our DP5K dataset. An illustration of it is presented in Fig. 3. To make this paper self-contained, we begin with a brief remark on the DP image formation model.

3.1 DP image formation mdoel

Image blur can be modeled as the convolution of a sharp image and a PSF [16]:

$$\mathbf{I}_b\left(i,j\right) = \mathbf{H} * \mathbf{I}_s + \eta, \quad \text{s.t.} \sum \mathbf{H} = 1, \tag{1}$$

where $*$ denotes convolution, $\mathbf{I}_b\left(i,j\right)$ is the pixel value of the blurred image $\mathbf{I}_b$ at position (i,j), $\mathbf{H}$ is the spatially-varying PSF, $\mathbf{I}_s$ is the sharp image patch centered at (i,j) with the same size to $\mathbf{H}$, and η denotes noise. In the context of DP sensors, their PSFs can record more information than the ordinary ones owing to the unique hardware design. Specifically, as shown in Fig. 4, a single DP pixel consists of two photodiodes, each of which can only receive photons passing through half of the lens [1]. Therefore, given a scene point out of the depth-of-field (DoF), the defocus blur on the two DP views will appear in opposite directions to form the so-called DP disparity effect, providing information about how relatively distanced a certain point is w.r.t. the DoF, whether a certain area on the image is blurred, and where the defocus blur occurs (*i.e*., in front of the DoF or behind it). Furthermore,



Minimum depth (mm)	Maximum depth (mm)	Average depth (mm)	Median depth (mm)
264	10000	3020	2769

Fig. 5: Statistics of our DP5K dataset. Top: Histogram of focus distances. Bottom: Statistics of depths.

assuming that the lens is symmetric, the left and right DP PSFs satisfies a vertical symmetry [3]. Consequently, Eq. (1) for DP sensors can be written as

$$\begin{aligned} \mathbf{I}_l(i,j) &= \mathbf{H}_l * \frac{\mathbf{I}_s}{2} + \eta_l, \\ \mathbf{I}_r(i,j) &= \mathbf{H}_r * \frac{\mathbf{I}_s}{2} + \eta_r, \\ \text{s.t.} \sum \mathbf{H}_l &= 1,\ \mathbf{H}_r = \text{hflip}(\mathbf{H}_l), \end{aligned} \tag{2}$$

where $\mathbf{I}_l$, $\mathbf{I}_r$, η_l, and η_r denote the left and right images and noises of the DP view, respectively, $\mathbf{H}_l$ is the left PSF, and $\text{hflip}(\cdot)$ is the pixel-wise left-right flipping operator.

3.2 Dataset summary

An optics-orientated DP simulator generally requires sharp images, depth maps, and imaging parameters used in Eq. (6) as inputs. Our dataset captures 513 indoor scenarios containing such information, resulting in 5130 pairs of sharp images, DP defocus blur images, depth maps, and imaging parameters. We process the data following the procedure mentioned above and divide them into 3850 pairs for training, 640 pairs for validation, and 640 pairs for test. Some statistics of our dataset are presented in Fig. 5.

Apart from the processed data, we also provide access to the raw files, each of which additionally contains the 10 blur and 2 sharp images in the raw format, 6 ChArUCo board images for calibration, and 92 gray-code images for structured-light.

3.3 Hardware setup

Although broad consumer devices have employed DP sensors nowadays, we are only aware of Google and Canon as the manufacturers that provide users access to raw DP images. To obtain high-quality images, we employ the Canon EOS 5D Mark IV cameras with the Canon EF 85mm f/1.8 USM lens to capture full-frame DP images.

To obtain accurate depth maps, we double the aforementioned camera and lens, and further employ an EPSON EB-U42 projector to construct a stereo structured-light system. The entire hardware setup is shown in Fig. 6.

Fig. 6: Our DP structured-light stereo setup.

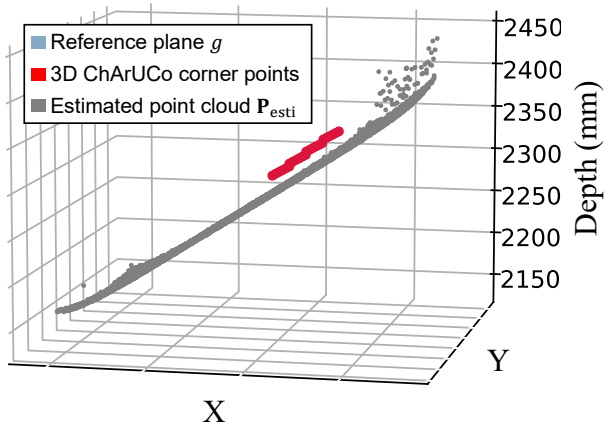


Fig. 7: An illustration of the reference plane and our estimated one.

3.4 Capture setup

We fix the focus point to a manually selected pixel and capture five blurred DP images with f-numbers set to $f/1.8$, $f/2$, $f/2.8$, $f/4$, and $f/5.6$, respectively. For the sharp ones, we follow Abuolaim and Brown [5] and set the f-number to $f/22$ to approximate a pinhole camera.

For acquiring depth maps with the structured-light stereo system, we employ the gray-code pattern [17] with positive-negative projections for better decoding accuracy. Furthermore, since changing the focus distance will lead to slight forward-backward movements of the lens, the calibration parameters also change accordingly. Therefore, we re-conduct the stereo calibration process by taking three images of a ChArUco board [18] whenever the focus point is changed.

3.5 Depth estimation & Accuracy evaluation

Here we present our depth estimation pipeline. Since depth map acts a critical role in the downstream CoC map generation and focus distance estimation, we also conduct an assessment of the accuracy of our depth acquisition pipeline.

Depth estimation: We use the 3DUNDERWORLD algorithm [19] to obtain depth maps from stereo structured-light images. We empirically find that the dark albedos on some of our photographed objects make the slim gray-code bars difficult to be detected, resulting in invalid depth values. Therefore, to obtain the depth

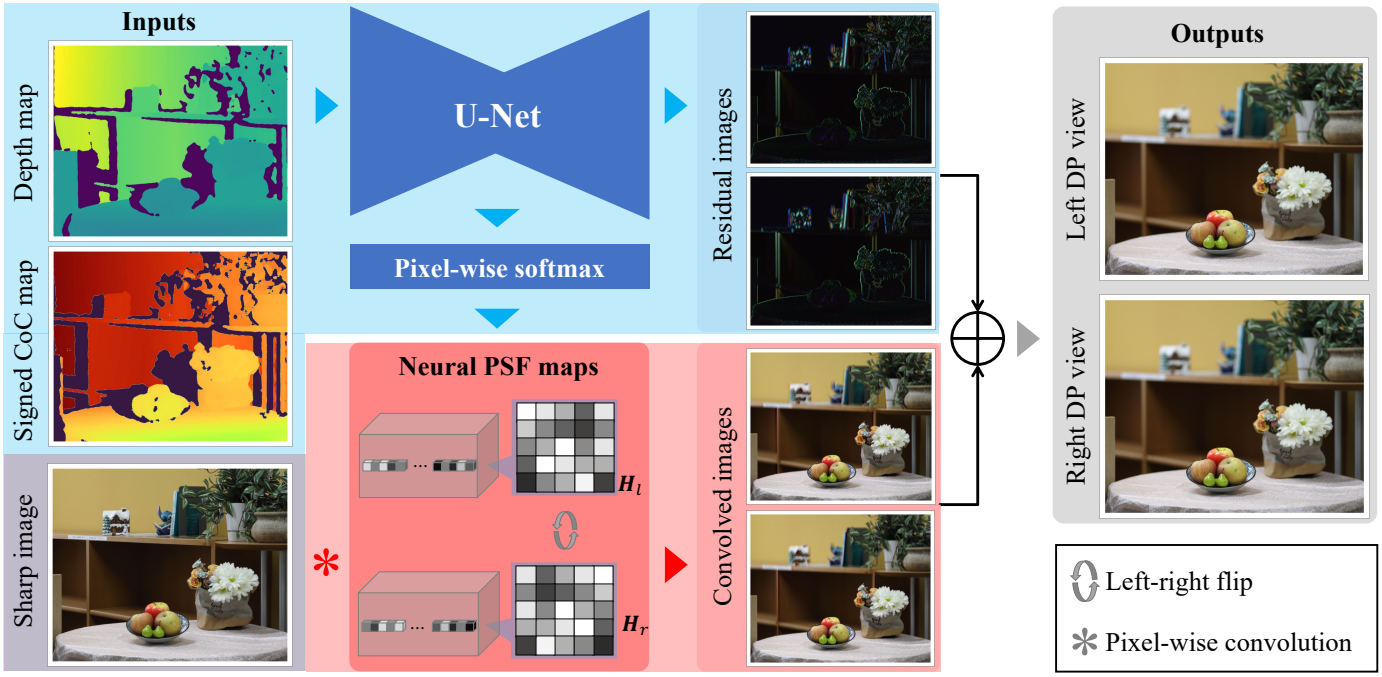


Fig. 8: Network structure of Neural DP Simulator. Blue: Inputting all the three images to the U-Net to obtain the residual images and the left pixel-wise neural PSF map. Red: Flipping the left neural PSF map to obtain the right one, and pixel-wise convolving them with the sharp image.

maps as dense as possible, we use an image pyramid over the resolution of the projector with scales 1, 1/2, 1/4, and 1/8, and decode their patterns in the observed images independently to obtain four depth maps. These depth maps are first fed into a morphological dilatation-erosion filter [20] to close small holes. Then, they are projected to the 3D space to apply a point cloud statistical filter [21], and back-projected to 2D to apply a median filter [22] over the valid depth pixels to remove outliers. Finally, we compute the depth map in a recursive manner:

$$\mathbf{D}_i = \mathbf{D}_{i-1} + \mathbf{M}_i \odot \mathbf{D}_{s_i}, \quad s_i \in \{1,\ 1/2,\ 1/4,\ 1/8\}, \quad (3)$$

where $\mathbf{D}_i$ is the final depth map in the i^{th} step with $\mathbf{D}_0$ being an all-zero matrix, $\odot$ is the Hadamard product, $\mathbf{M}_i$ is a binary mask indicating the invalid depth pixels of $\mathbf{D}_{i-1}$, and $\mathbf{D}_{s_i}$ is the filtered depth map with projector resolution scale s_i. The step i begins from 1 and the iteration is terminated after the process of $i = 3$. Finally, the obtained depth map is unrectified to match the coordinates of the DP images.

Accuracy evaluation of acquisted depth map: we assess the accuracy of our depth acquisition pipeline using ChArUCo markers [18]. Specifically, we print a paper of ChArUCo markers, paste it on a wall approximately 2.2 meters away from the structured light system, and estimate a 3D point cloud $\mathbf{P}_{\text{esti}}$ of the wall with our structured-light system. Then, we compare the point cloud $\mathbf{P}_{\text{esti}}$ with the pose and distance of the wall, which are represented by a plane equation g obtained by the detected markers, as shown in Fig. 7. From these data, a relative error can be calculated as

$$\text{relative err} = \frac{\text{mean}\left(\text{PtP}\right)}{\text{mean}\left(\mathbf{D}_{\text{esti}}\right)}, \quad (4)$$

where PtP denotes the set of point-to-plane distances between point cloud $\mathbf{P}_{\text{esti}}$ and plane g, and $\mathbf{D}_{\text{esti}}$ is the depth map corresponding to $\mathbf{P}_{\text{esti}}$. Following Eq. (4), the relative error of our structured-light system is 0.59%, which is comparably accurate to commercial RGBD cameras (for reference, the relative error of the commercial RealSense D415 RGBD camera[1] is 2%).

3.6 Focus distance estimation

In order to obtain the focus distance F of each capture, we first extract the focus point coordinates from the EXIF data, and use it as the center to create a window $\mathbf{W}$ of size $(30, 20)$ on the depth map. Then, we apply the Huber-skip robust estimator [23] over this window to reject possible outliers and use the remained ones to compute a robust mean value, which we record as the focus distance. The overall computation can be written in the form of

$$F = \text{mean}\left(\left|\frac{\mathbf{W}\left(i,j\right) - \text{median}\left(\mathbf{W}\right)}{\text{MAD}\left(\mathbf{W}\right)}\right| \leq \tau\right), \quad (5)$$

where $\text{mean}(|\cdot| \leq \tau)$ is the mean of all the entries that are bounded by τ, MAD is the median absolute deviation, $\mathbf{W}\left(i,j\right)$ indicates the depth value at coordinate (i,j), and τ is a threshold consistently set to 3.5 throughout the dataset generation.

3.7 DP view acquisition

We extract DP views from the raw files using Digital Photo Professional[2]. All the images, including the depth maps, are finally resized to $(1680, 1120)$ and saved as 16-bit lossless images. More information w.r.t. our data-processing pipeline is presented in the supplementary material.

1. RealSense D415 https://www.intelrealsense.com/depth-camera-d415/, last accessed on July 7, 2023.
2. Digital Photo Professional https://id.canon/en/support/0200583602, last accessed on April 14, 2023.

4 Neural DP Simulator

Owing to the large-scale dataset mentioned above, in this work, we also come up with a data-driven DP Simulator that can convert off-the-shelf pinhole RGBD datasets to DP counterparts. Compared to existing simulators relying on handcrafted models, our data-driven proposal can parameterize the DP imaging model more precisely in an implicit manner.

4.1 Network design of the Neural DP Simulator

We base the network structure of our simulator on the kernel prediction networks [24] to imitate the DP image formation model presented in Eq. (2). Figure 8 illustrates an overview of our network. Specifically, we employ a U-Net [25] with a pixel-wise softmax layer to imitate the properties of the pixel-wise PSFs, leading to a map of flattened pixel-wise convolution kernels (*i.e.*, a 3D tensor), which we call the flattened pixel-wise left neural PSF maps. Then, we pixel-wise reshape this left neural PSF maps to 2D, left-right flip it to obtain the right neural PSF maps, and convolve these two neural PSF maps with the sharp image to emulate the image formations shown in Eq. (2). Also, since real-world PSFs can be significantly large in size and hence requiring a large receptive field, to avoid huge memory consumption, we constrain the size of the predicted pixel-wise convolution kernels to 5×5 and let the U-Net additionally predict two residual images. These two images are added to the convolution results to jointly compensate for the restricted receptive fields and the noises η mentioned in Eq. (2).

The inputs of the simulator requires careful consideration. Specifically, although a sharp image and a signed CoC map are sufficient for DP image synthesis, they cannot characterize the positions of invalid depth pixels, which unavoidably exists in real-world RGBD frames. This is because all numerical values, including the negatives, zero, and the positives, represent meaningful CoC radii, leading to the fact that we cannot set any indicating values solely on the CoC maps. To solve such a problem, we design our Neural DP Simulator to jointly take a sharp image, a signed CoC map, and a depth map as inputs. The depth map is normalized to $(0, 1)$ with invalid depth values set to -1, aiming to help the network to identify invalid pixels. To obtain the signed CoC map, we follow [26] and compute its pixel values in the form of

$$\mathbf{C}\left(i,j\right) = \frac{f_{\text{thin}}^2}{Nd} \cdot \frac{\left(d - F\right)}{\left(F - f_{\text{thin}}\right)}, \tag{6}$$

where F is the focus distance, f_{thin} is the focal length in the context of the thin-lens model, N is the f-number, and d is the object distance (*i.e.*, absolute depth). In the inference phase, users can arbitrarily control the rendered DP defocus blur effects by modifying the inputted CoC map.

4.2 Generating CoC maps for off-the-shelf pinhole RGBD datasets

Existing RGBD datasets are rich resources for DP image synthesis if a trained Neural DP Simulator is available. However, they are chiefly captured with cameras that assume the pinhole model, and hence only provide the intrinsic parameters that cannot be directly fed to Eq. (6) for CoC computation. Therefore, we here describe the method for computing CoC maps for such datasets.

While the users can flexibly define the focus distances and the f-numbers, the thin-lens focal lengths f_{thin} require careful treatment. Specifically, although the intrinsic parameters in the existing RGBD datasets have already contained a "focal length" term, these values (hereafter denoted as f_{pin}) are under the assumption of the pinhole model, and hence different from f_{thin}.

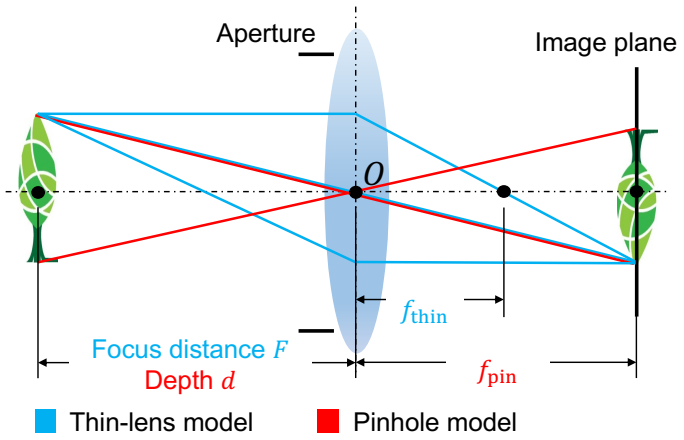


Fig. 9: Relation between the thin-lens and the pinhole models.

Let us first remark the definitions of f_{thin} and f_{pin}. As shown in Fig. 9, f_{thin} stands for the distance between the optical center and the on-axis point where refracted incoming parallel lights (*i.e.*, focused on infinity) intersect. On the other hand, f_{pin} is defined as the distance between the optical center (*i.e.*, the pinhole) and the image plane. Furthermore, since the aperture size does not affect the focus distance, given an in-focus scene point, the pinhole model can be converted from a thin-lens one by narrowing down the aperture size to infinitesimal, and the resulting f_{pin} is exactly the image distance of the thin-lens model. Accordingly, the thin-lens equation can be written as

$$\frac{1}{f_{\text{thin}}} = \frac{1}{F} + \frac{1}{f_{\text{pin}}}, \tag{7}$$

which allows to convert f_{pin} to f_{thin} given the focus distances F.

5 Experiments

In this section, we compare our DP simulator with the existing ones and study its effectiveness in training data generation. In all of the following experiments, we employ the Adam optimizer [27] and the cosine-annealing scheduler [28] for optimization, and select the checkpoints with the highest structure similarity (SSIM) scores on the validation sets for tests.

5.1 Training our Neural DP Simulator

We train our simulator on our collected DP5K dataset using a masked version of the edge-aware image similarity loss [29]:

$$\begin{aligned}\text{loss} = &D\left(\check{\mathbf{I}}_l, \hat{\mathbf{I}}_l, \mathbf{M}\right) + D\left(\nabla\check{\mathbf{I}}_l, \nabla\hat{\mathbf{I}}_l, \mathbf{M}\right) \\ &+ D\left(\check{\mathbf{I}}_r, \hat{\mathbf{I}}_r, \mathbf{M}\right) + D\left(\nabla\check{\mathbf{I}}_r, \nabla\hat{\mathbf{I}}_r, \mathbf{M}\right),\end{aligned} \tag{8}$$

where $\check{\mathbf{I}}_{(\cdot)}$ and $\hat{\mathbf{I}}_{(\cdot)}$ are the predicted and ground-truth DP views, $\mathbf{M}$ is a binary mask with 1 corresponding to the valid depth pixels and 0 to the invalid ones, $\nabla\left(\cdot\right)$ is the gradient operator, and $D\left(\cdot\right)$ is the masked version of the Charbonnier loss [30] in the form of

$$D\left(\check{\mathbf{I}}, \hat{\mathbf{I}}, \mathbf{M}\right) = \frac{\sum_{i,j}\sqrt{\mathbf{M}\left(i,j\right)\cdot\left(\check{\mathbf{I}}\left(i,j\right) - \hat{\mathbf{I}}\left(i,j\right)\right)^2 + \epsilon^2}}{\sum_{i,j}\mathbf{M}\left(i,j\right)}, \tag{9}$$

where ϵ is the Charbonnier parameter. We iterate the training process for 100 epochs. Other details are presented in the supplementary material.

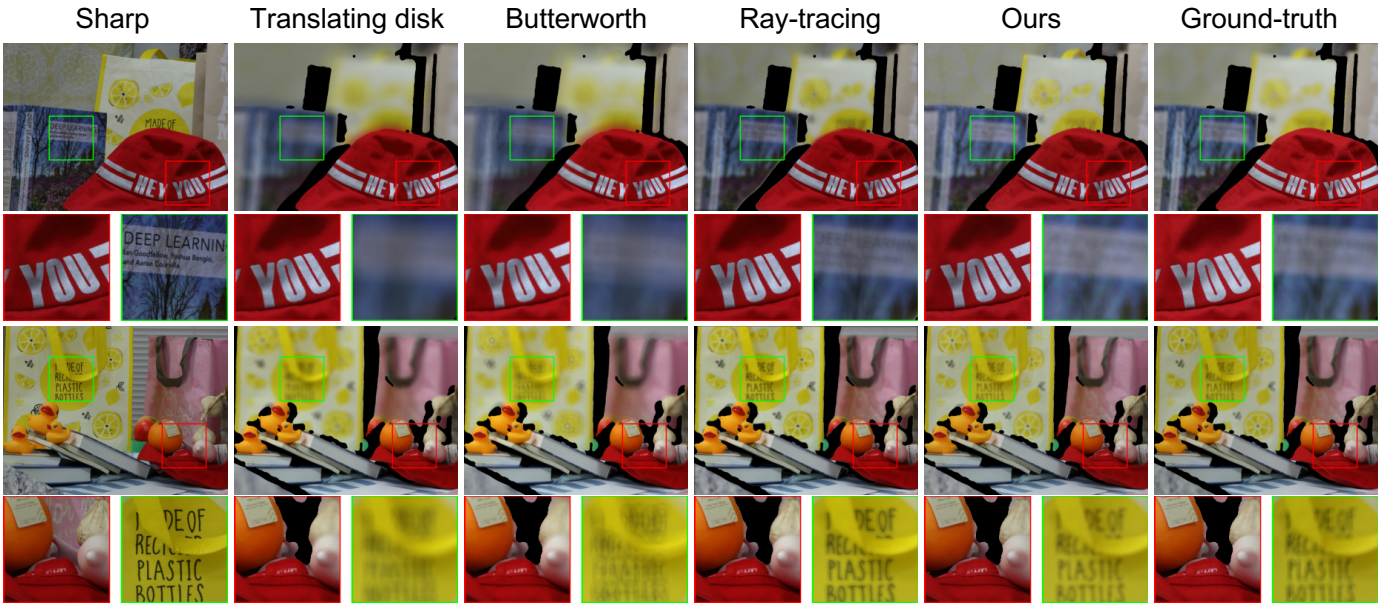


Fig. 10: DP images generated by different simulators with f-number $f/1.8$. All images are shown in the combination form. Pixels with unknown depths are painted black.

5.2 Comparing different DP simulators in terms of synthesizing photorealistic DP images

In this section, we conduct experiments to compare the accuracies of the DP images synthesized by our simulator and existing state-of-the-art methods.

5.2.1 Peer methods

We compare our Neural DP simulator with the existing ones that use the translating disk-based PSF [3], the Butterworth filter-based PSF [8], and ray-tracing [9], respectively. To let the former two approaches demonstrate their full capacities, we set their PSF sizes to the CoC radii. Also, for the translating disk-based method, instead of conducting a simple pixel-wise convolution between the PSF and the sharp image, we modify it to use the same discretization-and-blending technique proposed by the Butterworth filter-based approach. For the ray-tracing algorithm, we re-implement it by ourselves since the author-released executable file is not directly applicable to our data. All the hyper-parameters are maintained to be the default ones across different methods.

5.2.2 Accuracies of the synthesized DP images

We study the abilities of different simulators using the test set of our DP5K dataset. Since all of the four methods require depth values as inputs, we use the masked versions of the mean absolute error (MAE), the peak signal-to-noise ratio (PSNR), and the SSIM metrics for evaluation, where pixels corresponding to unknown depths are ignored.

Some examples generated by these simulators are shown in Fig. 10. Compared with other methods, the blurry image regions generated by our simulator are qualitatively closer to the ground truths. The quantitative results are shown in Table 2. Our simulator achieves the best accuracy in all metrics.

5.3 Benefits of our simulator in real-world tasks: A case study by defocus deblurring

DP defocus deblurring networks take blurred DP views as input and output the corresponding sharp images. To quantify the results, we continue using the standard MAE, PSNR, and SSIM metrics.

TABLE 2: Comparison of different DP simulators in synthesizing photorealistic DP images.

	Translating disk [3]	Butterworth [8]	Ray-tracing [9]	Neural DP Simulator (Ours)
MAE	0.049	0.052	0.055	**0.036**
PSNR	25.64	24.38	25.68	**30.23**
SSIM	0.85	0.78	0.90	**0.92**

TABLE 3: Defocus deblurring results on our DP5K dataset with data synthesized by different simulators.

	Translating disk [3]	Butterworth [8]	Ray-tracing [9]	Neural DP Simulator (Ours)
MAE	0.062	0.062	0.065	**0.047**
PSNR	23.27	23.22	22.76	**24.32**
SSIM	**0.82**	0.81	0.77	0.81

5.3.1 Generating synthetic DP images with our simulator

We manually select 4722 RGBD frames from the computer-generated Hypersim RGBD dataset [32] to generate the training data. All of the selected images are of resolution $(1024, 768)$ and depicting indoor scenes. For each frame, we randomly select a window of size $(60, 40)$ over the depth map as the focus area, and use its mean depth value as the focus distance. The f-number is randomly selected from $\{f/1.8, f/2, f/2.8, f/4, f/5.6\}$. For the thin-lens focal length f_{thin}, since the Hypersim dataset provides focus distances in millimeter unit and pinhole focal lengths in pixel unit, it is difficult to directly apply Eq. (7). We thus manually adjust the sensor size for each individual frame to ensure that the resulting CoC radii are within the same range as the training data. Some examples are illustrated in the supplementary material.

5.3.2 Defocus deblurring with data synthesized by different DP simulators

Fig. 11: Deblurring results generated by the MPRNet [29] trained on both synthetic and real-world data.

TABLE 4: Defocus deblurring results on our DP5K dataset with distinct sources of training data.

	DPDNet [5]			SRNNet [31]			MPRNet [29]		
	MAE	PSNR	SSIM	MAE	PSNR	SSIM	MAE	PSNR	SSIM
Our synthesized data only	0.047	24.32	0.81	0.040	27.21	0.83	0.046	25.69	0.81
A few real-world data only	0.087	18.44	0.67	0.041	26.93	0.82	0.038	27.72	0.85
Our synthesized data & A few real-world data	**0.041**	**25.76**	**0.84**	**0.037**	**27.81**	**0.85**	**0.036**	**28.74**	**0.87**
Reference: Entire DP5K dataset	0.039	26.04	0.86	0.034	28.39	0.87	0.028	30.58	0.92

TABLE 5: Ablation study of the network structure of our Neural DP Simulator.

	w/o residual images w/ neural PSF maps	w/ residual images w/o neural PSF maps	w/o residual images w/o neural PSF maps	w/ residual images w/ neural PSF maps
Masked MAE	0.0566	0.0368	0.0391	**0.0363**
Masked PSNR	26.062	29.831	29.040	**30.227**
Masked SSIM	0.874	0.914	0.909	**0.925**

We first study the relative effectiveness of different DP simulators in generating training data for defocus deblurring. For experimental setup, we use the three peer simulators mentioned in Section 5.2 to the same selected RGBD frames mentioned in the previous paragraph with the same imaging parameters. For each synthetic dataset individually generated by each of the four DP simulators including ours, the dual-pixel defocus deblurring network (DPDNet) [5] is trained for 50 epochs. The validation and test are carried out using our DP5K dataset. Other hyper-parameters are maintained to be the same among different trials and detailed in the supplementary material.

Table 3 shows the results. Although the translating disk-based method achieves a slightly higher score in SSIM, our simulator outperforms the others in MAE and PSNR with a large margin.

5.3.3 *Our Neural DP Simulator for transfer learning*

Transfer learning is a reliable approach for combining synthetic and real-world training data to improve performance [33], and it is especially beneficial when the real-world data is on a small scale. To assess the effectiveness of our synthesized DP images in such a scenario, we set up the following three training settings:

(1) Only training with DP images synthesized by our simulator.
(2) Only training with a small number of real-world DP images.
(3) Pre-training with our synthesized DP images and fine-tuning with a small number of real-world ones.

As a reference, we also conduct training with our DP5K dataset (called Reference hereafter). For Settings (1), (2), and Reference, we set the number of epochs to 50, 100, and 100, respectively. For Setting (3), we use 50 epochs for pre-training and 50 for fine-tuning. Other setups are detailed in the supplementary material.

For a small-scale real-world dataset, we collect 57 (*i.e.*, 570 sharp-depth-defocus pairs) new training data following the same setup mentioned in Section 3, and use our DP5K dataset as the validation and test sets. For the deblurring networks, we employ the DPDNet [5], the SRNNet [31], and the MPRNet [29] and modify the latter two to take the 6-channel concatenated DP images as inputs. All these networks are equipped with their original losses except for the metric function, which is replaced with the Charbonnier loss for better performance. Some example results are depicted in Fig. 11. The quantitative results are summarized in Table 4. It is observable that the DP images synthesized by our Neural DP Simulator contribute to these networks to improve their accuracies, leading to closer results to Reference, where a large number of real-world data is used.

TABLE 6: Defocus deblurring results on the DPDD dataset. DPDD-part and DPDD-all denote the first 1000 and the entire 7000 training patches, respectively. Best results in the synthetic-only setup are underlined, and **bolded** in the synthetic & real-world one.

		Indoor			Outdoor			Indoor & Outdoor		
		MAE	PSNR	SSIM	MAE	PSNR	SSIM	MAE	PSNR	SSIM
Synth only	Translating disk [3]	0.043	24.39	0.81	0.062	21.30	0.67	0.053	22.81	0.73
	Butterworth [8]	0.044	24.42	0.80	0.062	21.35	0.66	0.053	22.84	0.73
	Ray-tracing [9]	0.046	24.23	0.77	0.065	21.01	0.62	0.055	22.58	0.69
	Ours	0.064	22.27	0.78	0.080	19.83	0.63	0.072	21.01	0.70
Synth & DPDD-part	Translating disk [3]	0.032	26.65	0.84	**0.053**	**22.75**	**0.72**	**0.043**	**24.65**	**0.78**
	Butterworth [8]	0.032	26.65	**0.85**	**0.053**	22.72	**0.72**	**0.043**	24.63	**0.78**
	Ray-tracing [9]	0.035	25.96	0.83	0.056	22.18	0.68	0.046	24.02	0.75
	Ours	**0.031**	**26.76**	0.84	0.055	22.46	0.68	**0.043**	24.59	0.76
Ref	DPDD-part only	0.061	20.73	0.71	0.082	18.85	0.58	0.072	19.77	0.64
	DPDD-all	0.026	28.31	0.87	0.052	23.14	0.74	0.039	25.66	0.80

5.4 Ablation study of the network structure

We conduct an ablation study of the network structure of our Neural DP Simulator on synthesis accuracy. Specifically, we alter the outputs of the U-Net to individually omit the estimations of the pixel-wise neural PSF maps, the residual images, and both of them (*i.e.*, let U-Net directly predict DP views) to assess the contributions of pixel-wise neural PSF maps and residual images. The training and test setups are maintained to be the same as mentioned in Sections 5.1 and 5.2. As shown in Table 5, our proposed network structure can lead to the best synthesis accuracy.

5.5 Generalization to different devices

It is preferred to train our Neural DP Simulator in a device-specific manner to achieve the highest synthesis accuracy. This is because, as a neural network-based approach, our generated DP images would naturally reflect the specific imaging sensor and lens models used for training data collection. Despite such a preference, we empirically find that our simulator remains beneficial if the real-world data captured with a different DP imaging system is insufficient. Specifically, we fine-tune the DPDNet pre-trained on our synthesized data using the first 1000 training patches of the DPDD dataset [5], which is captured by the same camera model as we used but several different lens models with the focus distance varying from 24 mm to 105 mm. The training setups are kept the same to the preceding paragraph. As shown in Table 6, our synthesized DP images can still improve the performances, leading to more similar results to the network trained with a large number of real-world data. This similarity is more visible on the indoor part as all the synthesized images depict indoor scenes. Moreover, our Neural DP Simulator can also achieve comparable accuracy w.r.t. other DP simulators when a little real-world data is available for fine-tuning. Therefore, we can conclude that, in terms of benefitting down-streaming tasks, the generalization ability of our work is not impeded by the device-specific property.

6 Conclusion and Discussion

This paper introduces a real-world DP dataset containing various types of information, hoping to benchmark and prompt DP simulator-related research. Based on this dataset, we also present Neural DP Simulator, which is a flexible tool to synthesize more photorealistic DP images from RGBD frames compared to state-of-the-art methods. Experiments show that our simulator can lead to effective data augmentation.

6.1 Limitation

The main limitation of our work lies in the trade-offs between the synthesis accuracy and the consumed time. Specifically, we prefer our simulator to be trained by device-specific data to better suit the device-specific property of real-world PSFs. As a side effect, our proposal requires repeating the data collection and network training process for each specific model of DP imaging systems. Despite its expensiveness for a single device, such a process is practically significant in some scenarios. For example, in smartphone manufacturing, one single trained simulator can suit millions of products that are equipped with the same DP imaging module. Moreover, data collection therein also becomes much easier owing to the integrated range sensors of modern smartphones. Also, as presented in the supplementary material, even if our simulator is trained with data captured by device A and directly applied to device B, the results appear to be comparable w.r.t. existing works.

6.2 Future works

We plan to use our simulator to generate data to train surface normal & albedo estimation networks, hoping to benefit successive applications such as relighting. Another thread of work lies in applying domain adaptation techniques to alleviate the device-specific data collection requirement.

Acknowledgments

Feiran Li would like to thank Ahmed Boudissa for his guidance in exploring dual-pixel sensors. This work was supported by JSPS KAKENHI Grant Number JP23H05491.

References

[1] H. Yukawa, "Solid-state imaging device and electronic apparatus," U.S. Patent 16 683 379, Nov. 2019. 1, 3

[2] R. Garg, N. Wadhwa, S. Ansari, and J. T. Barron, "Learning single camera depth estimation using dual-pixels," in *Proceedings of International Conference on Computer Vision (ICCV)*, 2019, pp. 7628–7637. 1, 2

[3] A. Punnappurath, A. Abuolaim, M. Afifi, and M. S. Brown, "Modeling defocus-disparity in dual-pixel sensors," in *Proceedings of International Conference on Computational Photography*, 2020, pp. 1–12. 1, 2, 4, 7, 9

[4] M. Kang, J. Choe, H. Ha, H.-G. Jeon, S. Im, and I. S. Kweon, "Facial depth and normal estimation using single dual-pixel camera," *arXiv preprint arXiv:2111.12928*, 2021. 1, 2

[5] A. Abuolaim and M. S. Brown, "Defocus deblurring using dual-pixel data," in *Proceedings of European Conference on Computer Vision (ECCV)*, 2020, pp. 111–126. 1, 2, 4, 8, 9

[6] A. Abuolaim, M. Afifi, and M. S. Brown, "Improving single-image defocus deblurring: How dual-pixel images help through multi-task learning," in *Proceedings of Winter Conference on Applications of Computer Vision (WACV)*, 2022, pp. 1231–1239. 1, 2

[7] S. Xin, N. Wadhwa, T. Xue, J. T. Barron, P. P. Srinivasan, J. Chen, I. Gkioulekas, and R. Garg, "Defocus map estimation and deblurring from a single dual-pixel image," in *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 2228–2238. 1, 2

[8] A. Abuolaim, M. Delbracio, D. Kelly, M. S. Brown, and P. Milanfar, "Learning to reduce defocus blur by realistically modeling dual-pixel data," in *Proceedings of International Conference on Computer Vision (ICCV)*, 2021, pp. 2289–2298. 1, 2, 7, 9

[9] L. Pan, S. Chowdhury, R. Hartley, M. Liu, H. Zhang, and H. Li, "Dual pixel exploration: Simultaneous depth estimation and image restoration," in *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 2, 7, 9

[10] Y. Zhang, N. Wadhwa, S. Orts-Escolano, C. Häne, S. Fanello, and R. Garg, "Du^{2}Net: Learning depth estimation from dual-cameras and dual-pixels," in *Proceedings of European Conference on Computer Vision (ECCV)*, 2020, pp. 582–598. 2

[11] A. Punnappurath and M. S. Brown, "Reflection removal using a dual-pixel sensor," in *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 1556–1565. 2

[12] X. Wu, J. Zhou, J. Liu, F. Ni, and H. Fan, "Single-shot face anti-spoofing for dual pixel camera," *Transactions on Information Forensics and Security*, vol. 16, pp. 1440–1451, 2020. 2

[13] P. P. Srinivasan, T. Wang, A. Sreelal, R. Ramamoorthi, and R. Ng, "Learning to synthesize a 4d rgbd light field from a single image," in *Proceedings of International Conference on Computer Vision (ICCV)*, 2017, pp. 2243–2251. 3

[14] T. Brooks and J. T. Barron, "Learning to synthesize motion blur," in *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 6840–6848. 3

[15] V. Sandfort, K. Yan, P. J. Pickhardt, and R. M. Summers, "Data augmentation using generative adversarial networks (cyclegan) to improve generalizability in ct segmentation tasks," *Scientific reports*, vol. 9, no. 1, pp. 1–9, 2019. 3

[16] K. R. Castleman, *Digital image processing*, 1996. 3

[17] H. Rubinsztein-Dunlop, A. Forbes, M. V. Berry, M. R. Dennis, D. L. Andrews, M. Mansuripur, C. Denz, C. Alpmann, P. Banzer, T. Bauer *et al.*, "Roadmap on structured light," *Journal of Optics*, vol. 19, no. 1, p. 013001, 2016. 4

[18] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*, 2008. 4, 5

[19] K. Herakleous and C. Poullis, "3dunderworld-sls: An open-source structured-light scanning system for rapid geometry acquisition," *arXiv preprint arXiv:1406.6595*, 2014. 4

[20] E. Dougherty, *Mathematical morphology in image processing*, 2018, vol. 1. 5

[21] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz, "Towards 3d point cloud based object maps for household environments," *Robotics and Autonomous Systems*, vol. 56, no. 11, pp. 927–941, 2008. 5

[22] T. Huang, G. Yang, and G. Tang, "A fast two-dimensional median filtering algorithm," *Transactions on Acoustics, Speech, and Signal Processing*, vol. 27, no. 1, pp. 13–18, 1979. 5

[23] P. J. Huber, "Robust estimation of a location parameter," in *Breakthroughs in statistics*, 1992, pp. 492–518. 5

[24] B. Mildenhall, J. T. Barron, J. Chen, D. Sharlet, R. Ng, and R. Carroll, "Burst denoising with kernel prediction networks," in *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 2502–2510. 6

[25] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proceedings of International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015, pp. 234–241. 6

[26] G. R. Fowles, *Introduction to modern optics*, 1989. 6

[27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2015. 6

[28] I. Loshchilov and F. Hutter, "SGDR: stochastic gradient descent with warm restarts," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2017. 6

[29] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, M.-H. Yang, and L. Shao, "Multi-stage progressive image restoration," in *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 14 821–14 831. 6, 8

[30] P. Charbonnier, L. Blanc-Feraud, G. Aubert, and M. Barlaud, "Two deterministic half-quadratic regularization algorithms for computed imaging," in *Proceedings of International Conference on Image Processing*, 1994. 6

[31] X. Tao, H. Gao, X. Shen, J. Wang, and J. Jia, "Scale-recurrent network for deep image deblurring," in *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 8174–8182. 8

[32] M. Roberts, J. Ramapuram, A. Ranjan, A. Kumar, M. A. Bautista, N. Paczan, R. Webb, and J. M. Susskind, "Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding," in *Proceedings of International Conference on Computer Vision (ICCV)*, 2021. 7

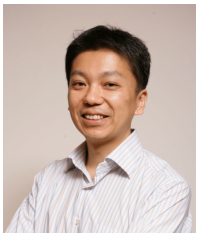
[33] L. Torrey and J. Shavlik, "Transfer learning," in *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, 2010, pp. 242–264. 8

Feiran Li received his B.Eng. degree from East China University of Science and Technology in 2017, M.Eng. degree from Nara Institute of Science and Technology in 2019, and Ph.D. degree from Osaka University in 2023, respectively. He is currently a Research Scientist with Sony Research. His research interest include computer vision and computation photography.

Heng Guo received his B.E. and M.S. degrees in signal and information processing from University of Electronic Science and Technology of China, and Ph.D. degree from Osaka University, in 2015, 2018, and 2022. He is currently a specially-appointed assistant professor at Osaka University. His research interests include physics-based vision and machine learning.

Hiroaki Santo received his M.S. and Ph.D. degrees in information science from Osaka University, Japan, in 2018 and 2021, respectively. He is currently an assistant professor with the Department of Multimedia Engineering, Graduate School of Information Science and Technology, Osaka University. His research interests include computer vision and machine learning.

Fumio Okura received his M.S. and Ph.D. degrees in engineering from the Nara Institute of Science and Technology in 2011 and 2014, respectively. He has been an Assistant Professor with the Institute of Scientific and Industrial Research, Osaka University, until 2020. He is now an Associate Professor with the Graduate School of Information Science and Technology, Osaka University. His research interest includes the boundary domain between computer vision and computer graphics.



Yasuyuki Matsushita received his B.S., M.S. and Ph.D. degrees in EECS from the University of Tokyo in 1998, 2000, and 2003, respectively. From April 2003 to March 2015, he was with Visual Computing group at Microsoft Research Asia. In April 2015, he joined Osaka University as a professor. His research area includes computer vision, machine learning, and optimization. He is an Editor-in-Chief of International Journal of Computer Vision (IJCV) and is/was on the editorial board of IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), The Visual Computer Journal, IPSJ Transactions on Computer Vision Applications (CVA), and Encyclopedia of Computer Vision. He served/is serving as a Program Co-Chair of PSIVT 2010, 3DIMPVT 2011, ACCV 2012, ICCV 2017, and a General Co-Chair for ACCV 2014 and ICCV 2021. He won the Osaka Science Prize in 2022. He is a senior member of IEEE and a member of IPSJ.