Simulating Dual-Pixel Images From Ray Tracing For Depth Estimation

Fengchen He, Dayang Zhao, Hao Xu, Tingwei Quan, Shaoqun Zeng
Huazhong University of Science and Technology
{linyark, dayangzhao, xuhao_2003, quantingwei, sqzeng}@hust.edu.cn

Abstract

Many studies utilize dual-pixel (DP) sensor phase information for various applications, such as depth estimation and deblurring. However, since DP image features are entirely determined by the camera hardware, DP-depth paired datasets are very scarce, especially when performing depth estimation on customized cameras. To overcome this, studies simulate DP images using ideal optical models. However, these simulations often violate real optical propagation laws, leading to poor generalization to real DP data. To address this, we investigate the domain gap between simulated and real DP data, and propose solutions using the Simulating DP Images from Ray Tracing (Sdirt) scheme. Sdirt generates realistic DP images via ray tracing and integrates them into the depth estimation training pipeline. Experimental results show that models trained with Sdirt-simulated images generalize better to real DP data. The code and collected datasets will be available at https://github.com/LinYark/Sdirt.
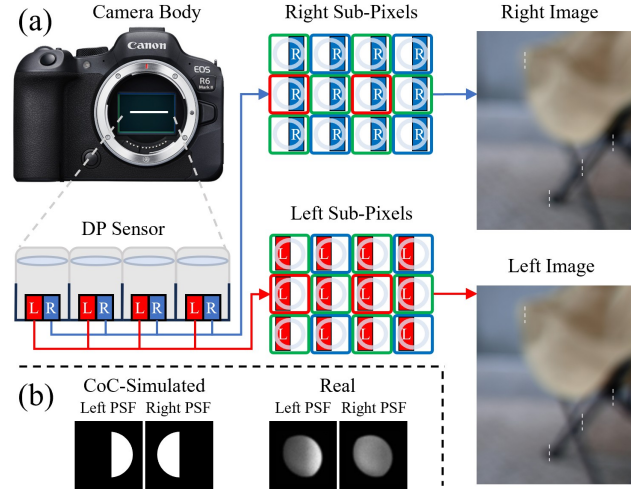
Figure 1. (a) Imaging process of a DP camera. The slight shifts between the left and right DP images caused by phase differences are illustrated with white dashed lines. (b) Example comparison between real and CoC-simulated DP PSFs, showing a significant difference between them.

1. Introduction

The DP sensor [15, 29] is designed to split each pixel into left and right sub-pixels, utilizing microlenses and sub-pixels to achieve phase splitting. This allows the capture of a pair of images in one shot, known as DP images, as shown in Fig. 1(a). DP images are not only used for autofocus [7, 11, 31], but also improve performance in tasks such as deblurring [4, 17, 41] and depth estimation [9, 13, 23]. In the field of depth-from-dual-pixel (DfDP) estimation, the scarcity of DP-depth paired datasets stems from hardware limitations. To address this, many studies [2, 16, 18, 23, 26, 37] have focused on simulating DP images using RGBD datasets, which serve as more flexible and accessible data sources.

The key to simulating DP images from RGBD datasets lies in accurately modeling the DP point spread function (PSF). In recent years, numerous model-based and calibration-based DP PSF simulators have been proposed. Calibration-based simulators [16, 18, 37] require substantial time calibrating real cameras to match DP PSFs, but suffer from issues such as interpolation errors due to discrete calibration points and the neglect of lens and DP sensor parameters, limiting model transferability. Model-based simulators [2, 23, 26] employ ideal optical or aberration analysis models to directly compute DP PSFs, bypassing calibration. However, these models often perform poorly on real DP images because they ignore lens aberrations and sensor phase splitting characteristics. As shown in Fig. 1(b), the ideal thin-lens model's simulated circle-of-confusion (CoC) DP PSF exhibits a significant domain gap with the real PSF, adversely affecting depth estimation.

To bridge the domain gap between real DP images and DP images simulated by model-based simulators, we propose Sdirt. Sdirt consists of a ray-traced DP PSF simulator and a pixel-wise DP image rendering module, which can accurately generate DP images with aberration and phase information. Specifically, given a fixed focus lens and DP sensor system with known parameters, we use ray trac-

ing [35, 39, 40] to calculate the spatially varying DP PSFs and train a multilayer perceptron (MLP) to predict them. Subsequently, we convolve per-pixel DP PSFs with the all-in-focus (AiF) image to simulate the DP images captured by a real camera. After training a DfDP model with DP images simulated by Sdirt, the model can accurately estimate depth using optical aberration and phase information, enhancing its ability to generalize to real DP images.

We conduct extensive experiments to validate the effectiveness of Sdirt. The experimental results demonstrate that, compared to other model-based simulators, the DP PSFs and DP images generated by Sdirt better resemble real data, and the DfDP model trained on Sdirt achieves superior generalization. In summary, our contributions are fourfold:

- We propose a ray-traced DP PSF simulator that computes spatially varying DP PSFs, addressing the domain gap between simulated and real DP PSFs caused by lens aberrations and sensor phase splitting.
- We propose a pixel-wise DP image rendering module that uses an MLP to predict the DP PSF for each pixel, narrowing the gap between simulated and real DP images.
- Depth estimation results show that the DfDP model trained on Sdirt generalizes better to real DP images.
- We collected DP119, a real DP-depth paired test set with an open lens structure and fixed focus, featuring diverse real-world scenes.

2. Related Work

2.1. DP Datasets

Most professional and mobile cameras have DP sensors, but only Canon cameras [1, 25, 26] and Google Pixel phones [9, 34, 43] offer DP data. In the field of depth estimation, supervised tasks rely on DP-depth paired datasets [1, 9, 26, 43]. However, changes in the lens, sensor, focal length, or aperture settings alter the PSFs, making it necessary to re-acquire DP-depth data. Consequently, many researchers focus on simulating DP-depth datasets using RGBD data to reduce data collection costs and improve flexibility.

2.2. DP PSF simulators

Simulating DP images from RGBD datasets relies on DP PSFs, with recent progress in calibration-based [16, 18, 37] and model-based [2, 23, 26] DP PSF simulators. In calibration-based simulators, Xin *et al*. [37] acquired full-space DP PSFs by sampling discrete points in space and interpolating, while Li *et al*. [18] acquired them through size scaling using the CoC model. Furthermore, Li *et al*. [16] generated DP PSFs using a U-Net, which requires a large amount of DP-depth paired data. Calibration-based simulators are time-consuming and prone to interpolation and scaling errors, while model-based simulators circumvent these issues. Pan *et al*. [23] simplified the optical system to an

ideal thin lens and used a symmetrically divided rectangular aperture to calculate DP PSFs. Punnappurath *et al*. [26] modeled the DP PSFs using a depletion kernel that is phase symmetric, meaning the left and right PSFs of the same object point are flip symmetric. Abuolaim *et al*. [2] used aberration analysis models to calculate DP PSFs, but the resulting patterns still remained phase symmetric. These studies oversimplify optical propagation by neglecting aberrations and phase splitting, leading to unrealistic domain gaps.

2.3. Applications driven by DP data

In recent years, DP data-driven applications such as depth estimation [9, 10, 13, 14, 23, 24, 43], deblurring [3, 4, 17, 27, 38, 41, 42], and refocusing [5, 7, 11, 32] have undergone rapid development. Additionally, Kang *et al*. [13] utilized DP data for facial scanning, similar to depth estimation tasks. Wadhwa *et al*. [34] employed DP data for synthetic shallow depth-of-field imaging, a popular recent application in mobile photography. Shi *et al*. [28] integrated DP hardware with structured diffractive optical elements for complex image encoding and high-precision multimodal reconstruction. In addition, DP data can be used for disparity estimation [22, 36], rain removal [18], and reflection removal [25]. It is foreseeable that DP data-driven applications will continue to emerge in various new fields.

3. Method of Sdirt

To provide the required input for training the DfDP network, Sdirt generates realistic DP images through ray tracing. As shown in Fig. 2(c), the module achieves pixel-wise rendering of DP images by convolving the DP PSF corresponding to each point in the depth map with the AiF RGB image. The DP PSF for each point is inferred by a pre-trained MLP network. As illustrated in Fig. 2(b), the offline training process of the MLP network uses ray-traced DP PSFs (Fig. 2(a)) as ground truth (GT).

3.1. Ray-traced DP PSF simulator

As shown in Fig. 2(a), lens ray tracing can accurately obtain the landing point on the sensor surface [8, 35, 39, 40]. We construct a ray set A containing n rays, each originating from an object point p. The aperture stop's image in object space is treated as the entrance pupil, from which n points are densely sampled. Each ray in A is initially directed from p to its corresponding entrance pupil sample. As rays in A traverse each lens surface, their positions and directions are updated by refraction according to Snell's law and the lens parameters. After multiple refractions, the rays reach the sensor plane. We denote the sensor-plane landing points as O and the corresponding ray directions as D.

DP sensor ray tracing is limited because camera manufacturers do not disclose the structural parameters of the microlens and sub-pixel components within the DP pixel.
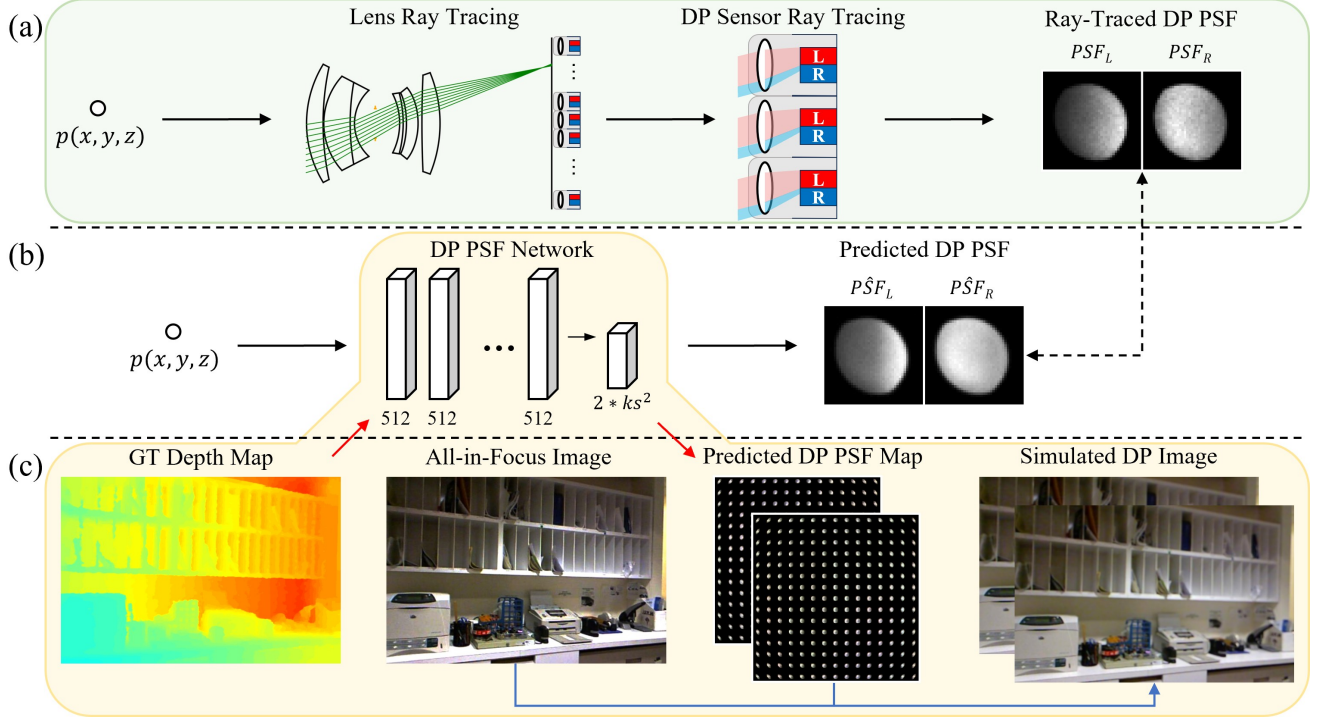
Figure 2. **Simulating Dual-Pixel Images from Ray Tracing pipeline.** (a) Ray-traced DP PSF simulator. Calculates spatially varying DP PSFs for lens and DP sensor through ray tracing. (b) DP PSF prediction network. Trains an MLP network to predict DP PSFs, using the ray-traced DP PSFs as GT. (c) Pixel-wise DP image rendering module. The network predicts the DP PSFs for all points in the depth map (red pass). Then, each DP PSF is convolved with the AiF RGB image to render the simulated DP image (blue pass).

As a result, we simplify the DP pixel structure (Fig. 3(a)) based on past research [15, 29]. We model the microlens as a thin lens with radius r and focal length f, and define h as distance between the sub-pixel and the microlens, w as sub-pixel width, and ps as DP pixel size.

The DP PSF can be obtained by separately calculating the cumulative integration of the rays for each sub-pixel. We analyze which sub-pixel the k-th ray A_k in the ray set A ultimately enters. Through lens ray tracing, we obtain the landing point $O_k(x_k, y_k, z_k)$ and direction $D_k(x'_k, y'_k, z'_k)$ of the k-th ray A_k on the sensor surface. Based on the landing point O_k, we can easily calculate that A_k lies within the DP pixel (i, j), which has surface center coordinates (x_i, y_i). For O_k, the situation differs depending on whether it lies **within** or **outside** the microlens:

When O_k lies within the microlens, as shown in Fig. 3(b), A_k is refracted by the microlens and then enters a sub-pixel. According to geometric optics theory, A_k and the principal ray incident on the thin lens at the same angle both converge on the focal plane of the thin lens. We only need to focus on a few boundary lines x_{L1}, x_{M1}, x_{R1} to determine which sub-pixel A_k ultimately enters. If x_k is in the interval $[x_{L1}, x_{M1}]$, it enters the left sub-pixel; if in the interval $[x_{M1}, x_{R1}]$, it enters the right sub-pixel; otherwise, it is considered a missing ray. The calculation methods for the

boundary lines are as follows:

$$
\begin{aligned}
x_{L1} &= x_i + w - (f * \tan\theta - w) * h/(f - h) \\
x_{M1} &= x_i - (f * \tan\theta) * h/(f - h) \\
x_{R1} &= x_i - w - (f * \tan\theta + w) * h/(f - h)
\end{aligned}
\tag{1}
$$

Where $\tan\theta$ is the tangent of the angle at which A_k lands on the sensor surface, equivalent to x'_k/z'_k. We provide the derivation hints for Eq. (1) in Fig. 4(a).

When O_k lies outside the microlens, as shown in Fig. 3(c), A_k enters the sub-pixel directly without refraction. Similarly, checking boundary lines x_{L2}, x_{M2}, x_{R2} suffices to determine which sub-pixel A_k ultimately enters:

$$
\begin{aligned}
x_{L2} &= x_i + w - h * \tan\theta \\
x_{M2} &= x_i - h * \tan\theta \\
x_{R2} &= x_i - w - h * \tan\theta
\end{aligned}
\tag{2}
$$

The left PSF (PSF_L) can be calculated by integrating the set of rays A emitted from point p over all left sub-pixels:

$$
PSF_L(i, j) = \sum_{k=1}^{n} A_k * \delta_{L,k}(i, j)
\tag{3}
$$

where $\delta_{L,k}(i, j)$ represents the energy distribution of ray A_k in the left sub-pixel at DP pixel (i, j). The ray energy distribution is assumed to be a unit impulse, which
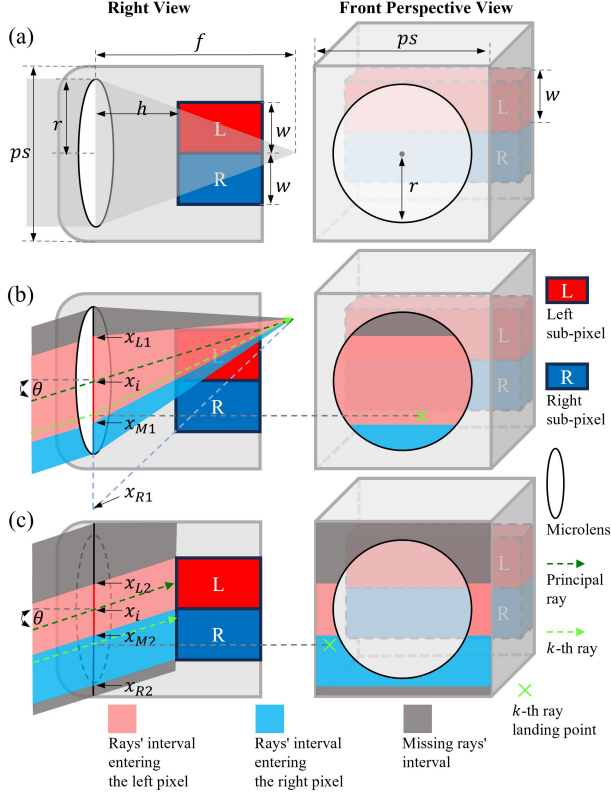
Figure 3. (a) DP pixel structure layout. The left part is a right-side view, and the right part is a front perspective view. (b) On hitting the microlens, the k-th ray lands on left/right sub-pixel in red/blue interval, else missed. (c) Without hitting the microlens, the k-th ray lands on left/right sub-pixel in red/blue interval, else missed.

equals zero unless it eventually enters the left sub-pixel of DP pixel (i, j). By following the same calculation steps as for PSF_L, PSF_R can also be computed.

3.2. Pixel-wise DP image rendering module

As shown in Fig. 2(b), we reduce the computational cost of ray-traced DP PSFs by training an MLP network to predict them, inspired by [33, 39]. The network input is the normalized coordinate p within the camera's valid imaging region (Fig. 4(b)), which is a frustum defined by the field of view, sensor size, and the preset minimum and maximum depths. We normalize the (x, y) coordinates to $[-1, 1]$. After fixing the focus distance, we set the DP PSF kernel size (ks) large enough to ensure that the DP PSF for any point within the valid imaging region can be fully displayed. The network consists of 5 hidden layers, each containing 512 neurons, and an output layer with $2 * ks^2$ neurons. We adopt the L2 loss to supervise predictions $P\hat{S}F_L$ and $P\hat{S}F_R$:

$$Loss = L_2(P\hat{S}F_L, PSF_L) + L_2(P\hat{S}F_R, PSF_R) \quad (4)$$

During training, we use ray-traced DP PSFs as GT and apply max normalization to alleviate learning difficulties caused by large radii PSFs. During inference, we apply
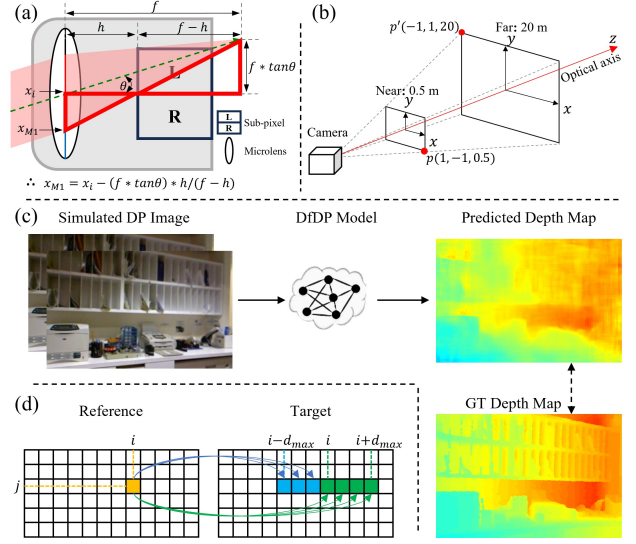

Figure 4. (a) Derivation hints for Eq. (1). (b) The valid imaging region is a frustum, and we normalize the (x, y) coordinates to $[-1, 1]$. (c) The DfDP model takes DP image as input to predict the depth map. (d) During cost volume generation in [6], we stack original disparity (green arrows) and add reverse disparity (blue arrows), with d_{max} as max displacement.

sum normalization to the predicted DP PSFs, approximating the uniform intensity distribution in cameras with vignetting compensation.

Because the DP PSF is shift-variant, we convolve each pixel's DP PSF with the AiF image to simulate the DP image captured by a camera. The RGBD dataset [21, 30] provides the paired RGB image (I_{RGB}) and depth map (I_D). As shown in Fig. 2(c), we treat each pixel in I_D as an object point and use the trained MLP to predict its DP PSF. Using I_{RGB} as the AiF image, we apply pixel-wise local convolution [39] to render the DP image containing aberration and phase information.

4. DfDP Model Based on Sdirt

As shown in Fig. 4(c), the input to the DfDP model is provided in real time by the pixel-wise DP image rendering module of Sdirt. At each training iteration, Sdirt feeds the DfDP model with simulated DP images of shape $(B, 6, H, W)$, where B is the batch size, the number of channels is 6, and H, W are the DP image height and width. The model outputs the predicted depth maps ($\hat{I}_D$) of shape $(B, 1, H, W)$. We apply the L1 loss to supervise $\hat{I}_D$:

$$Loss = L_1(\hat{I}_D, I_D) \quad (5)$$

We select [6] as the DfDP model and make reasonable adjustments to its cost volume step to accommodate the DP data. In binocular images, disparity from points at different depths always has the same direction, thus the cost volume generation process in [6] is also unidirectional, as shown by
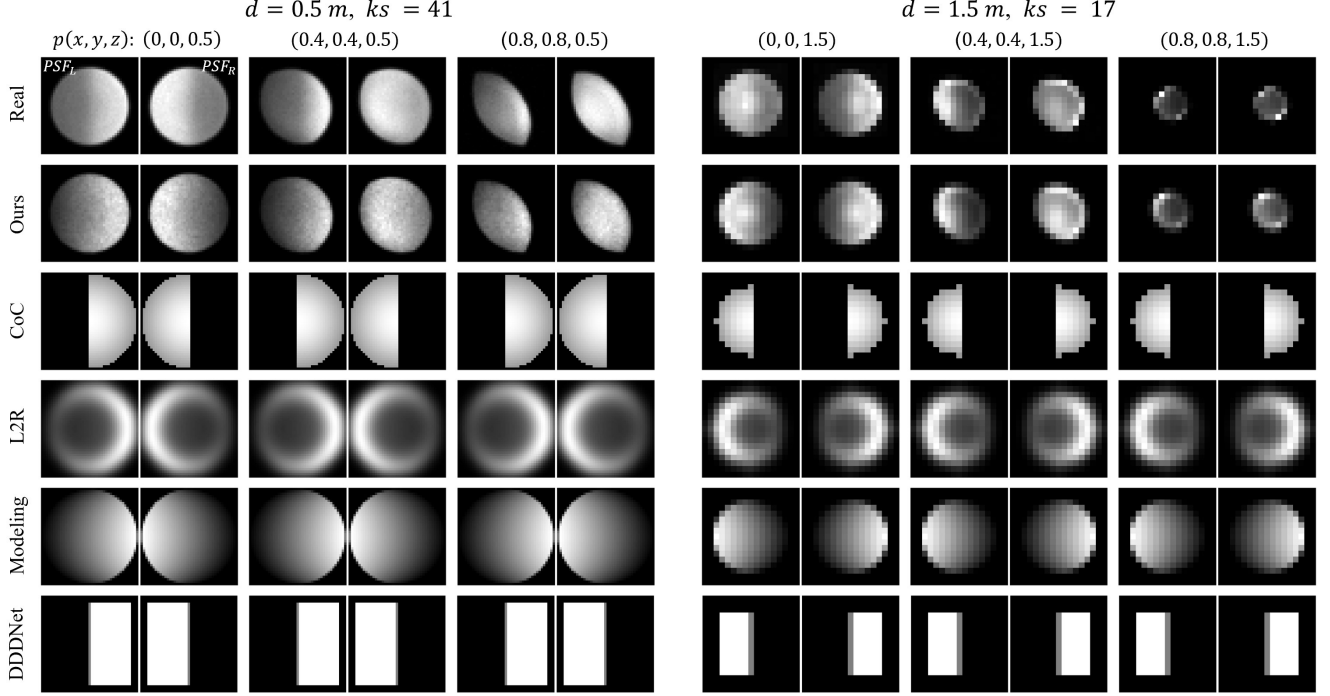
Figure 5. **Qualitative results of simulated DP PSFs.** Evaluate the real and simulated (ours, CoC, L2R [2], Modeling [26], and DDDNet [23]) F/1.8 DP PSFs at two depths (0.5 m and 1.5 m) and three different positions. As the object point p moves further from the optical axis, the real PSF_L and PSF_R become more phase asymmetric, and aberrations increase. Existing simulators neglect aberrations and DP phase splitting, causing a large gap between simulated and real DP PSFs. Only our ray-traced simulator predicts realistic results at all depths and positions.

the green arrows in Fig. 4(d). However, in DP images, disparity from object points before and after the focus distance is opposite in direction. Therefore, we add blue arrows to extend the cost volume generation, in order to enhance the DfDP model's ability to capture reverse disparity.

5. Sdirt Representation Result

5.1. Implementation details

Sdirt employs a Canon RF50mm lens and an R6 Mark II camera body as the real camera. The simulated DP sensor has dimensions of $24\,\mathrm{mm} \times 36\,\mathrm{mm}$ with a resolution of 512×768. We set the focus distance of both the simulated and real cameras to 1 m, the valid imaging range to 0.5–20 m, the F-number to F/4, and the ray set A to contain 4096 rays. During MLP training, ks is set to 21 to fully display the DP PSF of any point within the valid imaging region. We train the MLP for 100,000 iterations, selecting 128 random points within the valid imaging region during each iteration. We conducted our experiments on a 12700K CPU and a 3090 GPU.

Following [39], we assume chromatic aberration is well corrected compared to other aberrations and defocus effects, allowing us to use a 550 nm wavelength to reduce ray tracing costs. For details on obtaining the structural parameters of the DP pixel, please refer to the Supplementary.

5.2. Comparison methods

For comparison, we select all existing model-based DP PSF simulators: L2R [2], Modeling [26], DDDNet [23], and an added CoC simulator. Although [24] is a more recent work, its simulator fully reuses DDDNet [23], so we do not include it for comparison. Calibration-based simulators and constraint-training methods are data-driven and require real DP-depth paired data, making them fundamentally different from model-based simulators, so no comparison is made. We directly used the source code of these DP PSF simulators [2, 26], and re-implemented the simulator [23] which only provided executable files. To ensure fairness, we set the focus distance, F-number, and sensor size in each simulator to be consistent with ours.

5.3. Evaluation on simulated DP PSFs

To facilitate observation of aberration and phase information, we temporarily set the F-numbers of both the simulated and real cameras to F/1.8. We capture a OLED screen displaying a bright pixel to obtain the real DP PSFs.

As shown in Fig. 5, we evaluate the DP PSF (PSF_L on the left, PSF_R on the right) of object points p at various depths and positions within the valid imaging region. Observing the real DP PSFs, it can be found that defocus (with focus distance of 1 m) causes a significant phase difference
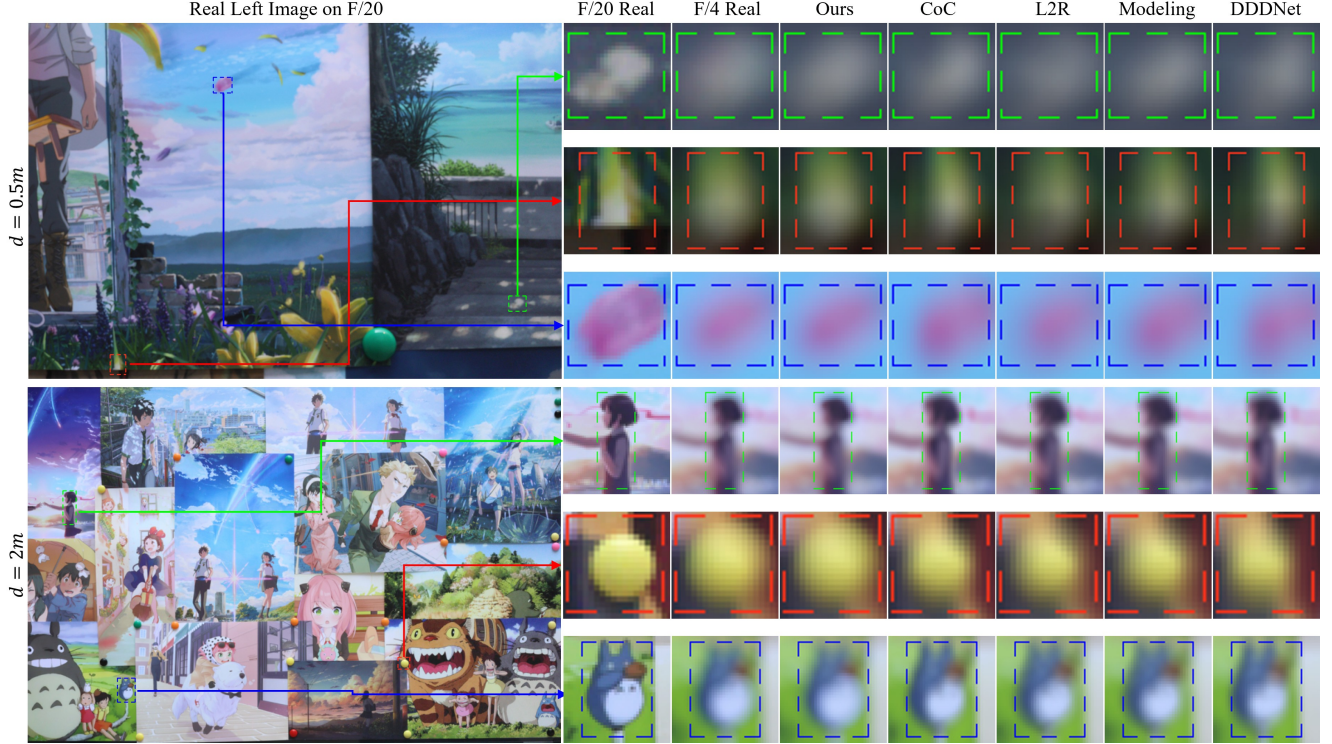
Figure 6. **Qualitative results of simulated DP images.** Evaluate the similarity between simulated (ours, CoC, L2R [2], Modeling [26], and DDDNet [23]) and real F/4 defocused left DP images at two depths (0.5 m and 2 m). Compared to real F/4 defocused images, images simulated by other methods exhibit varying sizes of patterns, incomplete shapes, and texture shifts in different directions before and after the focus distance (1 m). Our method produces the most realistic simulated images.

between the real PSF_L and PSF_R, and the phase difference at depths of 0.5 m and 1.5 m is opposite. When p is on the optical axis, all other simulators fail to provide DP PSF results close to the real ones. Farther from the optical axis, the real PSF_L and PSF_R become phase asymmetric, with smaller phase differences and more off-axis aberrations. The differences between DP PSFs from other simulators and the real ones become more apparent. In contrast, our ray-traced simulator predicts accurate results at all depths and positions.

We present the quantitative results in Tab. 1, using OpenCV's normalized squared difference (NSD) and normalized cross-correlation (NCC) to quantify the error and similarity between the real and simulated DP PSFs. We sampled 50 DP PSFs at 0.5 m and 1.5 m within the valid imaging region. Since the real DP PSF has flip symmetry (not phase symmetry) about both the x and y axes, we performed uniform sampling in the first quadrant. Our method achieves the highest similarity (0.915 NCC) and lowest error (0.133 NSD), demonstrating its effectiveness in bridging the domain gap between simulated and real DP PSFs introduced by lens aberrations and sensor phase splitting.

In all subsequent experiments, the F-number is adjusted back to F/4 due to the large blur kernel size caused by defocus at F/1.8, which resulted in GPU memory shortages.

Table 1. **Quantitative results of simulated DP PSFs.** The NCC and NSD metrics between simulated and real DP PSFs at 50 points were evaluated. Our method achieved the highest similarity and lowest error.

Method	Ours	CoC	L2R [2]	Modeling [26]	DDDNet [23]
NCC↑	**0.915**	0.672	0.638	0.707	0.589
NSD↓	**0.133**	0.448	0.523	0.423	0.625

5.4. Evaluation on simulated DP images

When comparing simulated and real DP images, to assess the degradation of simulated DP images caused by defocus effects at different depths, we captured richly textured planar scenes from 0.5 m to infinity, perpendicular to the optical axis. Shallower depths were sampled more densely to better capture defocus variations. This resulted in 56 scenes with varying depths, including 11 infinite-depth scenes. For each scene, real DP images were captured at F/4 and F/20. The F/20 image served as the AiF RGB image, and its capture-time depth was assigned to the entire depth map. The F/20 RGB-depth pairs were then input into DP simulators to generate the simulated F/4 defocused DP images.

As shown in Fig. 6, we compare the qualitative results of the defocused left DP images at depths of 0.5 m and 2 m. Due to errors in simulating the DP PSF, other methods produce simulated images with textures shifted to the right at
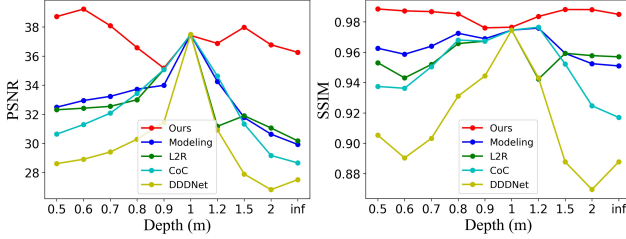
Figure 7. Quantitative results of simulated DP images. Evaluate the similarity of 56 real and simulated (ours, CoC, L2R [2], Modeling [26], and DDDNet [23]) planar scene F/4 DP images at different depths using PSNR (↑) and SSIM (↑) metrics. As the depth of simulated images deviates further from the focus distance (1 m), the less realistic the other methods become, whereas our scheme maintains the highest accuracy across all depths.

0.5 m depth and to the left at 2 m depth, compared to the real left DP image. Moreover, other methods show variations in image texture sizes and incomplete shapes. However, the DP images simulated by our Sdirt method show minimal deviation in texture details.

As shown in Fig. 7, which presents the quantitative results, we assess the similarity between simulated and real images using PSNR and SSIM. As the simulated image's depth moves further away from the focus distance (1 m), the neglect of aberrations and phase splitting in other methods leads to a more significant decrease in their metrics. In contrast, our Sdirt method achieves the best results at all depths, with average PSNR/SSIM values of 37.1982/0.9845, indicating that our simulated DP images are highly realistic.

6. DfDP Estimation Results

6.1. Implementation details

To train the DfDP model, we simulate F/4 DP images using Sdirt on RGBD datasets [21, 30]. The NYU Depth dataset [30], preprocessed by [12], contains 50,688 indoor scenes. We adopt the AdamW optimizer [19] and CosineAnnealing scheduler [20] with an initial learning rate of 1×10^{-4}. The model is trained with a batch size of 4 for 50 epochs. In each epoch, 2,000 randomly selected DP-depth pairs are used for training. After training, the model is directly evaluated on a real-world test set without any fine-tuning to assess its generalization ability.

6.2. A new real-world test set DP119

Unlike previous simulation methods, Sdirt addresses data scarcity by simulating hardware, requiring consistent camera and setup during training and inference. However, as shown in Tab. 2, no public dataset satisfies all of the following conditions: 1. Real DP-depth paired data for simulating DP images; 2. A known lens structure for ray tracing; 3. A fixed focus distance, since changing it alters the DP PSF of the same object point. This makes evaluation on public datasets with unknown hardware impractical. We there-

fore collected the DP119 test set, which contains diverse test scenes.

Table 2. Summary of existing DP datasets.

	DPDD [1]	L2R [2]	DPNet [9]	DP5K [16]	DDDNet [23]	Modeling [26]	DP119 Ours
Real captured	✓	×	✓	✓	×	✓	✓
Paired depth	×	✓	✓	✓	✓	×	✓
Lens structure	✓	-	×	✓	-	×	✓
Fixed focus dist.	✓	-	×	×	-	×	✓

DP119 consists of 45 planar scenes, 44 box scenes, and 30 casual scenes, totaling 119 scenes. The casual scenes represent common scenarios, suitable for evaluating the robustness of simulation models. The planar and box scenes are richly textured, helping the model utilize aberration and phase cues for depth estimation. Textureless areas, even when defocused, lack aberrations or phase differences, making depth estimation difficult and evaluation unreliable. Thus, richly textured planar and box scenes are ideal for assessing domain gaps between simulated and real DP images. Real DP images are captured by the Canon RF50mm lens with a R6 Mark II at F/4 and 1 m focus. GT depth maps for planar scenes are created by the depths at the capture time, while those for box and casual scenes are obtained from LiDAR scans. For more information about the DP119 test set, please refer to the Supplementary.

6.3. Evaluation

We present qualitative and quantitative results for the DfDP model trained by CoC, L2R [2], Modeling [26], DDDNet [23], and our Sdirt. For a fair comparison, we deploy their DfDP models using their DP PSF simulator source code [2, 26] and re-implemented code [23], and have all models share the same depth estimation network architecture, initialize from a common CoC pre-trained checkpoint, and follow the same implementation details.

Figure 8 shows depth estimation results for four casual scenes, with a color bar in meters for visualization. It can be seen that although other methods provide some relative positional information in the center region, they fail to correctly estimate relative positional relationships in the edge regions, and both the center and edge regions have large absolute positional errors. This is because the DP PSFs predicted by these methods exhibit phase symmetry and shift-invariance, whereas real DP PSFs exhibit phase asymmetry, aberrations, and shift-variance. Their methods' mismatch with real DP PSFs leads to significant depth estimation errors, especially near the edges. The results of L2R [2] and Modeling [26] are similar because the peak positions of their DP PSFs are quite close. In contrast, our method provides accurate relative positional information across the entire image area with minimal absolute positional errors, as our simulated DP PSFs are highly consistent with the real ones in both the center and edge regions.
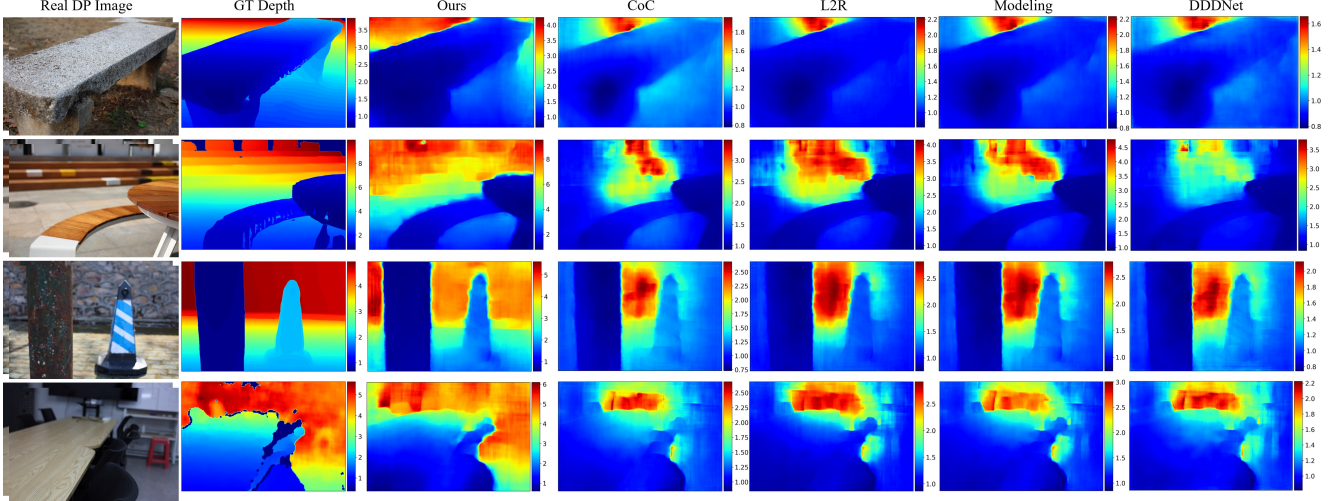
Figure 8. **Qualitative results of absolute depth estimation on DP119.** Evaluate DfDP models with CoC, L2R [2], Modeling [26], DDDNet [23] and our Sdirt at four casual scenes. Each result image includes a color bar showing depth in meters. Their depth estimation results show partial accuracy in relative positional relationships but large absolute positional errors. Our depth estimation results, however, demonstrate high accuracy in both relative and absolute positions with minimal errors. (Best viewed in color and enlarge on screen.)

Table 3. **Quantitative results of absolute depth estimation on DP119.** The DfDP model trained with our Sdirt achieves superior performance across most scenarios and evaluation metrics.

Scene	Method	MAE↓	MSE↓	Abs.r.↓	Sq.r.↓	Acc-1↑	Acc-2↑
	Ours	**0.0845**	**0.0109**	**0.0871**	**0.0095**	**0.9849**	**0.9997**
	CoC	0.2085	0.1001	0.1801	0.0659	0.6670	0.8990
Planar	L2R [2]	0.2418	0.1271	0.2112	0.0841	0.6319	0.8536
	Modeling [26]	0.2284	0.1142	0.2004	0.0766	0.6496	0.8725
	DDDNet [23]	0.2583	0.1485	0.2191	0.0958	0.5648	0.8089
	Ours	**0.1197**	**0.0339**	**0.0906**	**0.0231**	**0.9474**	**0.9812**
	CoC	0.3375	0.1804	0.2442	0.1116	0.4412	0.8277
Box	L2R [2]	0.3866	0.2284	0.2803	0.1412	0.3651	0.7156
	Modeling [26]	0.3655	0.2055	0.2660	0.1278	0.3907	0.7758
	DDDNet [23]	0.4177	0.2676	0.2975	0.1636	0.3456	0.6274
	Ours	**0.2702**	**0.2294**	**0.4632**	0.7241	**0.8236**	**0.9314**
	CoC	0.7925	1.8579	0.5461	0.6821	0.3318	0.6103
Casual	L2R [2]	0.8170	1.7487	0.5597	0.6719	0.2760	0.5315
	Modeling [26]	0.7934	1.7256	0.5510	**0.6655**	0.2978	0.5732
	DDDNet [23]	0.8931	2.0624	0.5752	0.7135	0.2481	0.4685

We evaluate all models on the DP119 test set using the following metrics to assess depth estimation performance: mean absolute error (MAE), mean squared error (MSE), absolute relative error (Abs.r.), squared relative error (Sq.r.), accuracy with $\delta < 1.25$ (Acc-1) and accuracy with $\delta < 1.25^2$ (Acc-2). Since the planar and box scenes are richly textured and do not have interference from textureless areas, the simulation models can rely on abundant aberration and phase difference cues for depth estimation. As shown in Tab. 3, in the simplest planar scene, our model achieves the best results across all metrics (0.9849 Acc-1). In the box scenes, all models experience a performance drop compared to the planar scenes, but we still achieve the optimal metrics (0.9474 Acc-1). These results demonstrate that our model is highly realistic and effectively bridges the domain gap between simulated and real DP images. Furthermore, we test the robustness of all models on the casual scenes. Due to

the textureless areas in the casual scenes, the performance of all models significantly degrades. However, our model still achieves the best metrics (0.8236 Acc-1), far surpassing the second-best model (0.3318 Acc-1). This further demonstrates that our model is the most realistic and has the best generalization performance. For more sample results from the DP119 test set, please refer to the Supplementary.

7. Conclusion and Discussion

Simulated DP images help address the scarcity of DP-depth paired data but face a domain gap between simulated and real DP data. In this work, we propose a novel simulation framework called Sdirt to bridge this gap. Specifically, we calculate the DP PSF for points in object space using ray tracing, and employ a network to predict them. Then, we render DP images based on the predicted DP PSFs. Experimental results show that the proposed Sdirt can simulate more realistic DP data. Moreover, depth estimation models trained based on Sdirt generalize better to real DP images.

We believe that Sdirt is not limited to DfDP tasks, it can provide extra depth information for any task with known optical imaging system parameters, promising significant applications in scenarios such as smartphones, automobiles, and microscopes in the future. However, Sdirt is only applicable to cameras equipped with a fixed-focus lens (the structure must be open) and a DP sensor (DP images must be available). Currently, only Canon (5D4, R series) meets these requirements. To further expand the application of Sdirt, more camera and smartphone manufacturers need to make these data accessible.

Acknowledgments

References

[1] Abdullah Abuolaim and Michael S Brown. Defocus deblurring using dual-pixel data. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*, pages 111–126. Springer, 2020. 2, 7

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

[3] Abdullah Abuolaim, Radu Timofte, and Michael S Brown. Ntire 2021 challenge for defocus deblurring using dual-pixel images: Methods and results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 578–587, 2021. 2

[4] Abdullah Abuolaim, Mahmoud Afifi, and Michael S Brown. Improving single-image defocus deblurring: How dual-pixel images help through multi-task learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1231–1239, 2022. 1, 2

[5] Hadi Alzayer, Abdullah Abuolaim, Leung Chun Chan, Yang Yang, Ying Chen Lou, Jia-Bin Huang, and Abhishek Kar. Dc2: Dual-camera defocus control by learning to refocus. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21488–21497, 2023. 2

[6] Xuelian Cheng, Yiran Zhong, Mehrtash Harandi, Yuchao Dai, Xiaojun Chang, Hongdong Li, Tom Drummond, and Zongyuan Ge. Hierarchical neural architecture search for deep stereo matching. *Advances in neural information processing systems*, 33:22158–22169, 2020. 4

[7] Myungsub Choi, Hana Lee, and Hyong-euk Lee. Exploring positional characteristics of dual-pixel data for camera autofocus. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13158–13168, 2023. 1, 2

[8] Geoffroi Côté, Fahim Mannan, Simon Thibault, Jean-François Lalonde, and Felix Heide. The differentiable lens: Compound lens search over glass surfaces and materials for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20803–20812, 2023. 2

[9] Rahul Garg, Neal Wadhwa, Sameer Ansari, and Jonathan T Barron. Learning single camera depth estimation using dual-pixels. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7628–7637, 2019. 1, 2, 7

[10] Bhargav Ghanekar, Salman Siddique Khan, Pranav Sharma, Shreyas Singh, Vivek Boominathan, Kaushik Mitra, and Ashok Veeraraghavan. Passive snapshot coded aperture dual-pixel rgb-d imaging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 25348–25357, 2024. 2

[11] Chi-Jui Ho, Chin-Cheng Chan, and Homer H Chen. Af-net: A convolutional neural network approach to phase detection autofocus. *IEEE Transactions on Image Processing*, 29:6386–6395, 2020. 1, 2

[12] Junjie Hu, Mete Ozay, Yan Zhang, and Takayuki Okatani. Revisiting single image depth estimation: Toward higher resolution maps with accurate object boundaries. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2019. 7

[13] Minjun Kang, Jaesung Choe, Hyowon Ha, Hae-Gon Jeon, Sunghoon Im, In So Kweon, and Kuk-Jin Yoon. Facial depth and normal estimation using single dual-pixel camera. In *European Conference on Computer Vision*, pages 181–200. Springer, 2022. 1, 2

[14] Donggun Kim, Hyeonjoong Jang, Inchul Kim, and Min H Kim. Spatio-focal bidirectional disparity estimation from a dual-pixel image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5023–5032, 2023. 2

[15] Masahiro Kobayashi, Michiko Johnson, Yoichi Wada, Hiromasa Tsuboi, Hideaki Takada, Kenji Togo, Takafumi Kishi, Hidekazu Takahashi, Takeshi Ichikawa, and Shunsuke Inoue. A low noise and high sensitivity image sensor with imaging and phase-difference detection af in all pixels. *ITE Transactions on Media Technology and Applications*, 4(2):123–128, 2016. 1, 3

[16] Feiran Li, Heng Guo, Hiroaki Santo, Fumio Okura, and Yasuyuki Matsushita. Learning to synthesize photorealistic dual-pixel images from rgbd frames. In *2023 IEEE International Conference on Computational Photography (ICCP)*, pages 1–11. IEEE, 2023. 1, 2, 7

[17] Yu Li, Yaling Yi, Dongwei Ren, Qince Li, and Wangmeng Zuo. Learning dual-pixel alignment for defocus deblurring. *arXiv preprint arXiv:2204.12105*, 2022. 1, 2

[18] Yizhou Li, Yusuke Monno, and Masatoshi Okutomi. Dual-pixel raindrop removal. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 1, 2

[19] I Loshchilov. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 7

[20] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 7

[21] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4040–4048, 2016. 4, 7

[22] Sagi Monin, Sagi Katz, and Georgios Evangelidis. Continuous cost aggregation for dual-pixel disparity extraction. In *2024 International Conference on 3D Vision (3DV)*, pages 675–684. IEEE, 2024. 2

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

[24] Liyuan Pan, Richard Hartley, Liu Liu, Zhiwei Xu, Shah Chowdhury, Yan Yang, Hongguang Zhang, Hongdong Li, and Miaomiao Liu. Weakly-supervised depth estimation and image deblurring via dual-pixel sensors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 2, 5

[25] Abhijith Punnappurath and Michael S Brown. Reflection removal using a dual-pixel sensor. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1556–1565, 2019. 2

[26] Abhijith Punnappurath, Abdullah Abuolaim, Mahmoud Afifi, and Michael S Brown. Modeling defocus-disparity in dual-pixel sensors. In *2020 IEEE International Conference on Computational Photography (ICCP)*, pages 1–12. IEEE, 2020. 1, 2, 5, 6, 7, 8

[27] Lingyan Ruan, Martin Bálint, Mojtaba Bemana, Krzysztof Wolski, Hans-Peter Seidel, Karol Myszkowski, and Bin Chen. Self-supervised video defocus deblurring with atlas learning. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024. 2

[28] Zheng Shi, Ilya Chugunov, Mario Bijelic, Geoffroi Côté, Jiwoon Yeom, Qiang Fu, Hadi Amata, Wolfgang Heidrich, and Felix Heide. Split-aperture 2-in-1 computational cameras. *ACM Transactions on Graphics (TOG)*, 43(4):1–19, 2024. 2

[29] Eun Sub Shim, Kyungho Lee, Junghyung Pyo, Wooseok Choi, Jungbin Yun, Taesub Jung, Kyungduck Lee, Seyoung Kim, Chanhee Lee, Seungki Baek, et al. All-directional dual pixel auto focus technology in cmos image sensors. In *2021 Symposium on VLSI Technology*, pages 1–2. IEEE, 2021. 1, 3

[30] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part V 12*, pages 746–760. Springer, 2012. 4, 7

[31] Przemysław Śliwiński and Paweł Wachel. A simple model for on-sensor phase-detection autofocusing algorithm. *Journal of Computer and Communications*, 1(06):11, 2013. 1

[32] Kunal Swami. Adjust your focus: Defocus deblurring from dual-pixel images using explicit multi-scale cross-correlation. In *International Conference on Computer Vision and Image Processing*, pages 318–330. Springer, 2023. 2

[33] Ethan Tseng, Ali Mosleh, Fahim Mannan, Karl St-Arnaud, Avinash Sharma, Yifan Peng, Alexander Braun, Derek Nowrouzezahrai, Jean-Francois Lalonde, and Felix Heide. Differentiable compound optics and processing pipeline optimization for end-to-end camera design. *ACM Transactions on Graphics (TOG)*, 40(2):1–19, 2021. 4

[34] 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 (ToG)*, 37(4):1–13, 2018. 2

[35] Congli Wang, Ni Chen, and Wolfgang Heidrich. do: A differentiable engine for deep lens design of computational imaging systems. *IEEE Transactions on Computational Imaging*, 8:905–916, 2022. 2

[36] Zhuofeng Wu, Doehyung Lee, Zihua Liu, Kazunori Yoshizaki, Yusuke Monno, and Masatoshi Okutomi. Disparity estimation using a quad-pixel sensor. *arXiv preprint arXiv:2409.00665*, 2024. 2

[37] Shumian Xin, Neal Wadhwa, Tianfan Xue, Jonathan T Barron, Pratul P Srinivasan, Jiawen Chen, Ioannis Gkioulekas, and Rahul Garg. Defocus map estimation and deblurring from a single dual-pixel image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2228–2238, 2021. 1, 2

[38] Hao Yang, Liyuan Pan, Yan Yang, Richard Hartley, and Miaomiao Liu. Ldp: Language-driven dual-pixel image defocus deblurring network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24078–24087, 2024. 2

[39] Xinge Yang, Qiang Fu, Mohamed Elhoseiny, and Wolfgang Heidrich. Aberration-aware depth-from-focus. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 2, 4, 5

[40] Xinge Yang, Qiang Fu, and Wolfgang Heidrich. Curriculum learning for ab initio deep learned refractive optics. *Nature communications*, 15(1):6572, 2024. 2

[41] Yan Yang, Liyuan Pan, Liu Liu, and Miaomiao Liu. K3dn: Disparity-aware kernel estimation for dual-pixel defocus deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13263–13272, 2023. 1, 2

[42] Dafeng Zhang and Xiaobing Wang. Dynamic multi-scale network for dual-pixel images defocus deblurring with transformer. In *2022 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2022. 2

[43] Yinda Zhang, Neal Wadhwa, Sergio Orts-Escolano, Christian Häne, Sean Fanello, and Rahul Garg. Du 2 net: Learning depth estimation from dual-cameras and dual-pixels. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 582–598. Springer, 2020. 2