Supplemental Material: Revisiting Disparity from Dual-Pixel Images: Physics-Informed Lightweight Depth Estimation

Teppei Kurita Yuhi Kondo Legong Sun Takayuki Sasaki Sho Nitta
Yasuhiro Hashimoto Yoshinori Muramatsu Yusuke Moriuchi
Sony Semiconductor Solutions Corporation
{Teppei.Kurita, Yuhi.Kondo, Legong.Sun, Takayuki.Sasaki, Sho.Nitta,
Yasuhiro.Hashimoto, Yoshinori.Muramatsu, Yusuke.Moriuchi}@sony.com
https://github.com/sony/dual-pixel-disparity

This supplement is outlined as follows. References to the main study (Section, Equations, Figures, and Tables) are highlighted in blue. Section 1 provides an overview of the dual-pixel (DP) simulator and details on its parameter settings as described in Sec. 3 and Sec. 4 of the main study. Section 2 describes the details of the network architecture in Sec. 4 and Sec. 5 of the main study. Section 3 details the experiments discussed in Sec. 5 of the main study, including our synthetic dataset, the experimental setup, additional experimental results, and assessment details. Section 4 discusses other detailed considerations.

1. Dual-pixel simulator

Figure 1 provides an overview of the DP simulator used in Sec. 3 and Sec. 4 of the main study. The simulator is based on the one developed by Abuolaim et al. [3]. We also utilize their parametric model of the point spread function (PSF) of DP. Using the ground truth depth and all-in-focus RGB images as inputs, we determine the kernel size of the DP's PSF according to the depth and then convolve it. The left and right PSFs are convolved with the image, respectively, to generate the left and right images of the DP with disparity.

More efficient and effective DP simulator implementation: To enhance the efficiency of the DP simulator [3], we have precomputed and stored pre-convolved defocus maps for each PSF size, which accelerates the simulation by approximately threefold. Additionally, while the DP simulator [3] is limited to odd PSF kernel sizes (e.g., 3, 5, 7), by blending PSFs of different sizes, we can apply any natural number of kernel sizes to achieve more realistic simulations.

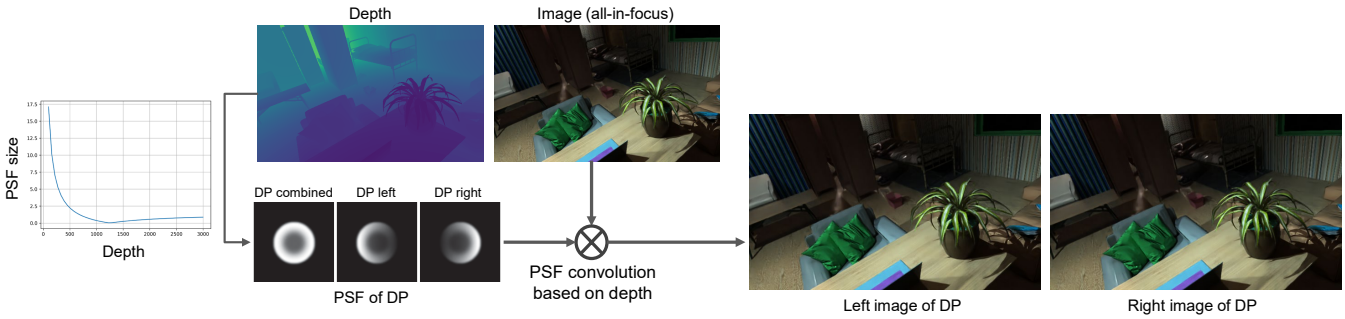


Figure 1. **Overview of dual-pixel simulator.** Using the ground truth depth and all-in-focus RGB images as inputs, we determine the kernel size of the DP's PSF according to the depth and then convolve it. The left and right PSFs are convolved with the image, respectively, to generate the left and right images of the DP with disparity.

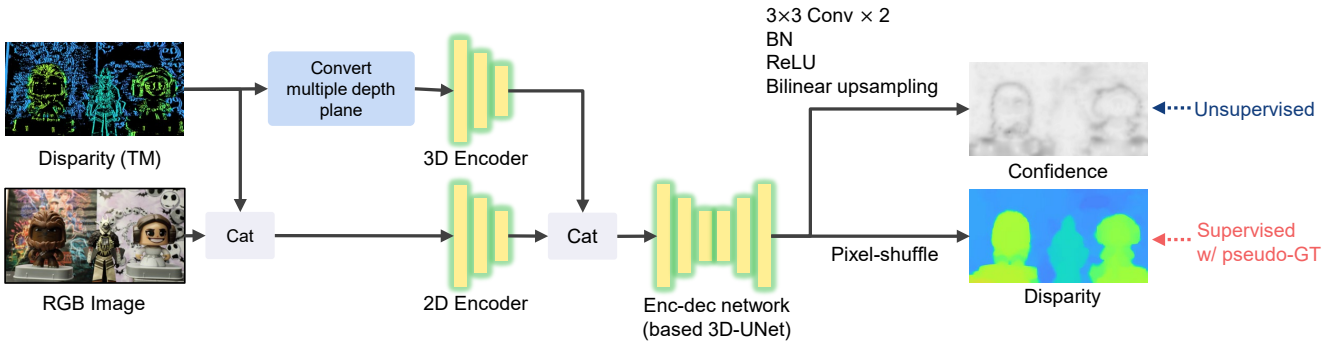


Figure 2. **Network architecture for outputting confidence maps based on CostDCNet.** The confidence map is output separately from the disparity by the encoder-decoder network.

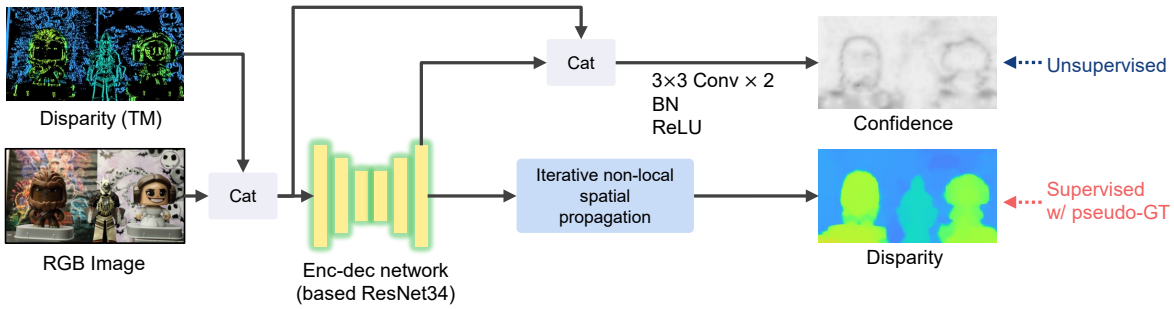


Figure 3. **Network architecture for outputting confidence maps based on NLSPN.** The features from the decoder's final layer are extracted and concatenated with the input RGB image and disparity. They are then convolved to output a confidence map.

Details of simulation in experiments: In the toy experiment shown in Fig. 4 of the main study, the simulator parameters are: depth is 2.76 m, focus distance is 0.94 m, $f/2.0$, and focal length is 50 mm. For calculating the template matching (TM) error of the DP image in Fig. 6 of the main study, the simulator parameters are: depth ranges from 0.1m to 3m (in 0.01m increments), focus distance is set from 0.6m, 0.8m, 1.0m, 1.2m, and 1.5m, f-number is set from 1.4, 2.0, 2.8, 4.0, and 5.0, and the focal length is 50 mm. In this case, the parameter α in Eqn. 1 of the main study is optimized using a golden section search to best match the disparity between the real-world and simulation data.

2. Network architecture

2.1. Network architecture for outputting confidence maps based on CostDCNet

This section provides details on the completion network used in Sec. 4 of the main study. An overview of the network is shown in Fig. 2. We have modified the network to output a confidence map based on CostDCNet [5], a lightweight completion network with few parameters. This network takes RGB and sparse disparity as inputs and outputs dense disparity and confidence maps. The sparse disparity is converted to a multiple depth plane and then input to the 3D encoder, while the RGB image is concatenated with the disparity and then input to the 2D encoder. Next, the feature outputs from the 3D and 2D encoders are concatenated and input to the encoder-decoder network, specifically the 3D-UNet. Finally, the features output from the 3D-UNet are pixel-shuffled to output disparity, convolved twice to output the confidence map, passed through batch normalization (BN) and ReLU, and upsampled. At this stage, the disparity is learned in a supervised manner, and the confidence map is learned unsupervised using the loss function described in Sec. 4, Eqn. 7 of the main study. Refer to [5] for a more detailed network configuration.

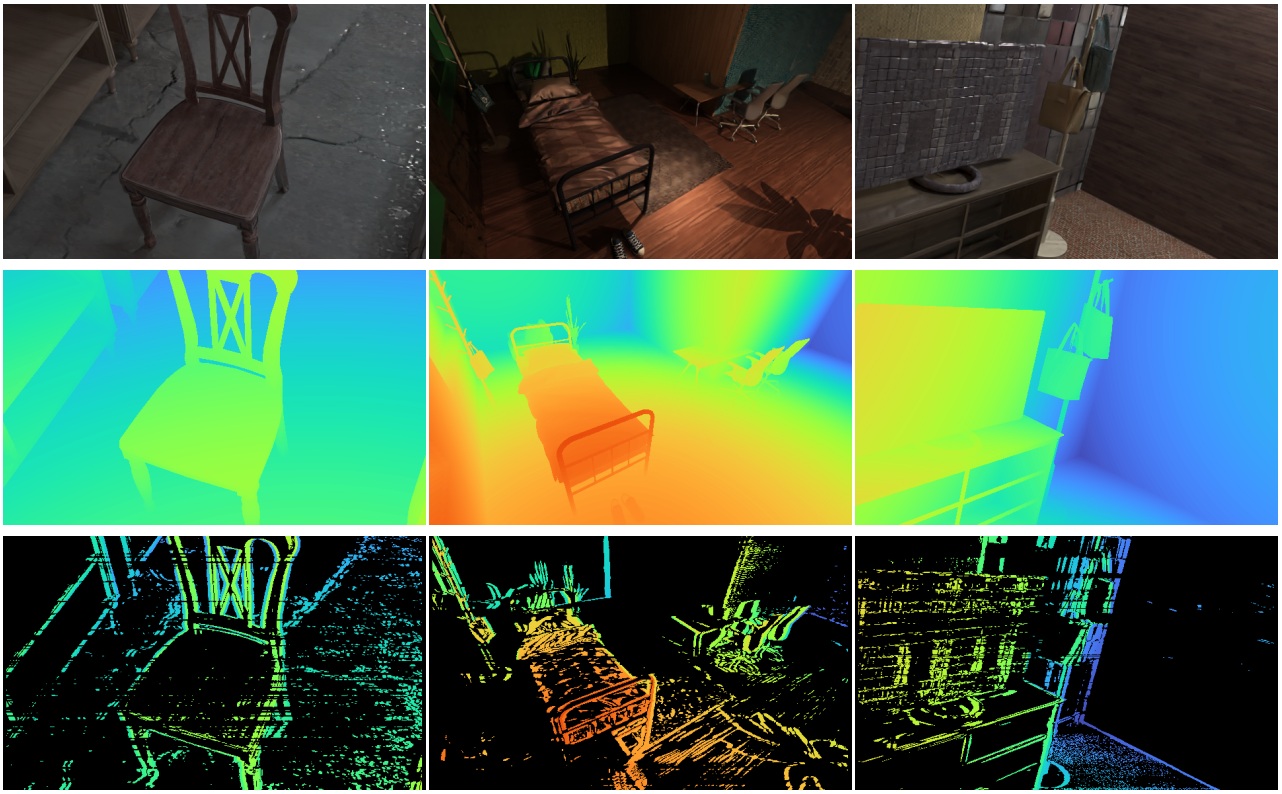


Figure 4. **Synthetic dataset generated by simulation.** The dataset consists of 10,000 samples, including RGB images (top row), ground truth depth (middle row), and edge depth (bottom row).

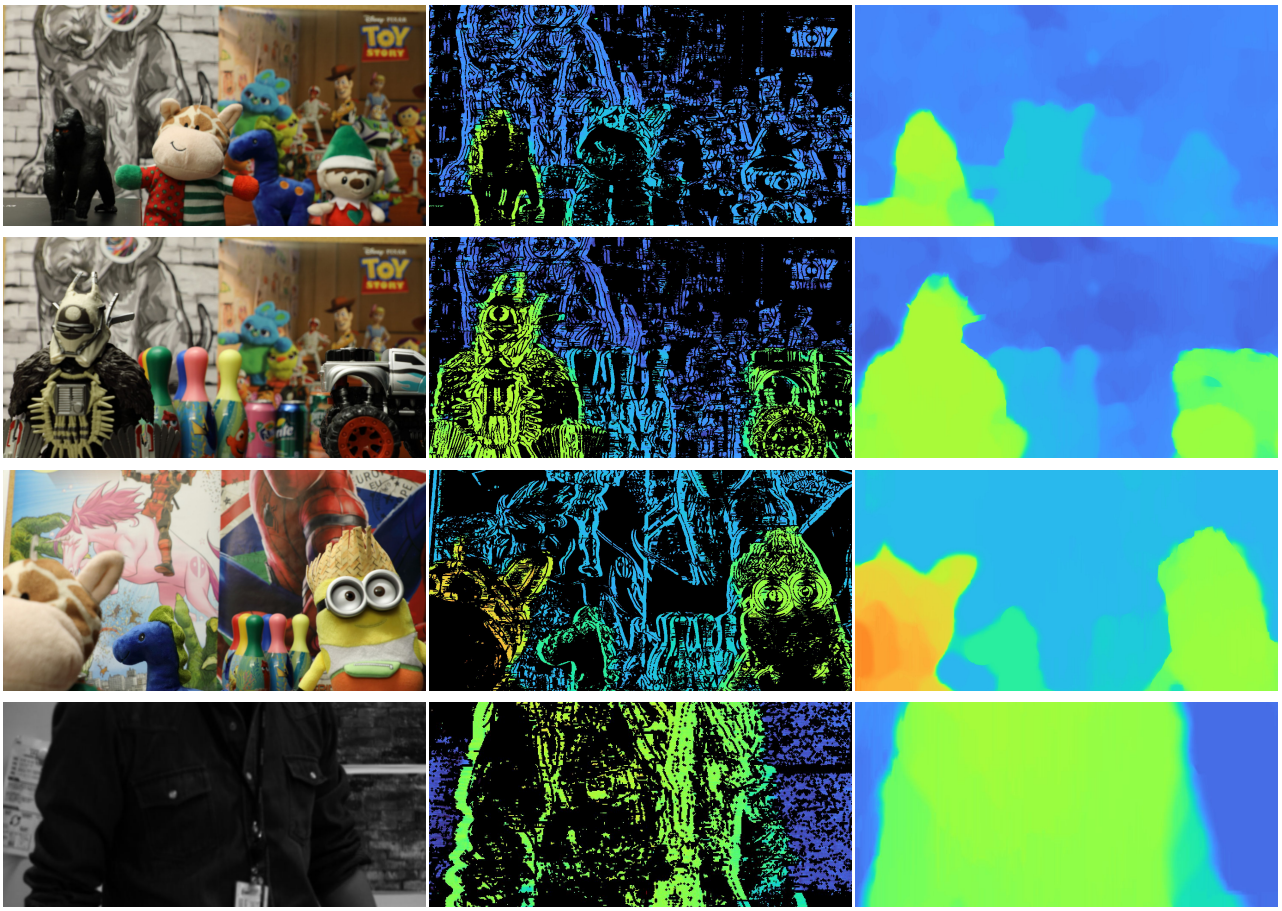
2.2. Network architecture for outputting confidence maps based on NLSPN

This section provides details on the completion network used in Sec. 5 of the main study. An overview of the network is presented in Fig. 3. The network is based on NLSPN [8], a completion network with more parameters and better performance than CostDCNet, modified to output a confidence map. Like CostDCNet, this network inputs an RGB image and sparse disparity and outputs dense disparity and confidence maps. The RGB image and sparse disparity are concatenated and fed into an encoder-decoder network based on ResNet34. The dense disparity is obtained by applying a recursive non-local spatial propagation filter to the features' output from the encoder-decoder network. To obtain a confidence map, the feature from the last layer of the decoder is concatenated with the input RGB image and disparity. Then, convolution is performed twice, followed by batch normalization and ReLU to obtain the confidence map. The loss function is the same as for CostDCNet. Please refer to [8] for a more detailed network configuration.

3. Experiment

3.1. Dataset details

This section provides details on the datasets in Sec. 5.1 of the main study. In addition to public RGBD datasets such as NYUDv2 [10], we developed a ray-tracing renderer to generate a synthetic RGBD dataset. This synthetic dataset consists of 10,000 samples of RGB images, depth, and edge depth sets, as shown in Fig. 4, at a resolution of 640×384. We used Houdini [1] to automatically generate objects and interior views, thus obtaining a large amount of data at minimal cost. We used a rule-based floor plan generation method [4]. The objects and camera were randomly positioned according to the rules, and textures and bump maps of objects were obtained from the Unreal Engine marketplace [2]. As in Sec. 4 of the main study, edge depth was obtained by performing edge detection on the RGB image, generating a mask, and multiplying the mask by the ground truth depth. This dataset enables efficient training in Sec. 4 of the main study by pre-computing edge depths. This dataset will be available on our project website in the future.



(a) Image (b) Disparity map (TM) (c) Estimated disparity map

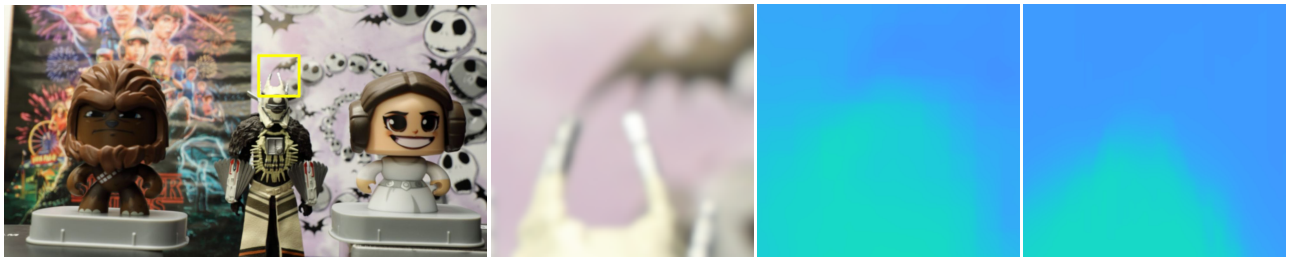
Figure 5. **Additional qualitative results.** Our method performs well without any major failures. In particular, our method does not generate false disparities for surfaces with uniform disparities.

Training using datasets with different characteristics: Each dataset has its unique characteristics and drawbacks. The real-world datasets have RGB images that are close to actual data, but occlusions and noise affect the quality of the ground truth depth. On the other hand, our synthetic dataset using computer graphics has no ground truth errors, but there is a domain gap with real-world scenes. Therefore, to prevent domain-specific training that includes these drawbacks as much as possible, we performed training by mixing and randomly sampling from three datasets with different characteristics (NYUDv2, RGB-D-D, our dataset).

3.2. Experimental setting details

This section provides details on the experimental setup in Sec. 5.1 of the main study. The resolution of the Punnappurath et al. dataset [9] used for evaluation was 5180×2940. This resolution was reduced to $1/16$ of its original size ($1/4$ height and $1/4$ width) to 1295×735 before template matching was performed. The edge disparity image obtained from template matching was cropped to 1184×648 and then input to the completion network. The disparity output from the network was enlarged 16-fold ($4\times$ height, $4\times$ width) to 2386×1296, then it was cropped to 2277×1188 for the final evaluation. The image cropping method was the same as that used in previous studies. The training of the completion network was performed at a resolution of 640×384.

Next, we provide details on the training of the completion network shown in Fig. 5 of the main study. The focus distance in the conversion from depth to disparity in Eqn. 1 of the main study is randomly set from 500mm to 3000mm in 1mm increments. The camera parameters are as follows: the focal length is 50mm, and the f-number is randomly selected from 1.4, 2.0, 2.8, 4.0, and 5.0. The parameter α is set randomly from 0.2 to 0.4.



(a) Scene (b) RGB image (c) Kim et al. [6] (d) Ours

Figure 6. **Close-up of results on thin, fine structures.** Due to the properties of the DP sensor, estimating disparity in such thin, fine structures is also challenging for conventional end-to-end methods. While our method shows improvements, there are still issues in achieving detailed reconstruction.

3.3. Additional results

Figure 5 shows results for other scenes that could not be included in the main study due to space constraints. Similar to the main study, the results were obtained using CostDCNet, a lightweight model, as a completion network. The bottom row shows the data we additionally obtained using a Canon DP camera (EOS 5D Mark IV). Our method performs well without any significant failures. In particular, our method does not generate false disparities on surfaces with uniform disparities.

Additionally, Fig. 6 shows close-up results of disparity estimation in thin, fine structures. Due to the properties of the DP sensor, estimating disparity in such thin, fine structures is challenging for conventional end-to-end methods as well. While our method shows improvements, there are still challenges in achieving detailed reconstruction. This remains a significant challenge for future research in disparity estimation using DP sensors.

3.4. Assessment details

Table 1 presents the quantitative evaluation results of the proposed method using a more complex completion network, NLSPN, as shown in Fig. 11 of Sec. 5.2 of the main study. Compared to the CostDCNet model used in the main study, the NLSPN is a more complex model with 26.4M parameters, which is more than 13 times larger. The results confirm that our proposed physics-informed learning method is effective even for more complex models. Furthermore, we have confirmed that our refinement framework is practical for such models and achieves state-of-the-art performance.

Table 1. Quantitative evaluation results of the proposed method when the completion network is a more complex model, NLSPN.

Configuration	AI(1)↓	AI(2)↓	$1-\lvert\rho_{\mathrm{s}}\rvert\downarrow$	Param
Wadhwa et al. [11]	0.0463	0.0740	0.2875	$\simeq 0$
Punnappurath et al. [9]	0.0449	0.0724	0.2301	$\simeq 0$
Pan et al. [7]	0.0894	0.1491	0.5008	11.0M
Kim et al. [6]	0.0390	0.0679	0.2092	10.6M
Ours, CostDCNet-based	0.0381	0.0704	0.1121	
Ours, CostDCNet-based w/ disparity refiner	0.0317	0.0684	0.0816	1.9M
Ours, CostDCNet-based w/ disparity and confidence refiner	0.0301	0.0667	0.0782	
Ours, NLSPN-based	0.0363	0.0701	0.1114	
Ours, NLSPN-based w/ disparity refiner	0.0315	0.0668	0.0787	26.4M
Ours, NLSPN-based w/ disparity and confidence refiner	**0.0296**	**0.0644**	**0.0741**	

4. Other technical details

4.1. Probability distribution function for sampling the error

Regarding the probability distribution function for sampling the error $n(\sigma_{\mathrm{d}})$ in Eqn. 5 of the main study: From the toy experiment in Fig. 4 of the main study, we observed that the error distribution of DP most closely fits a Laplace distribution.

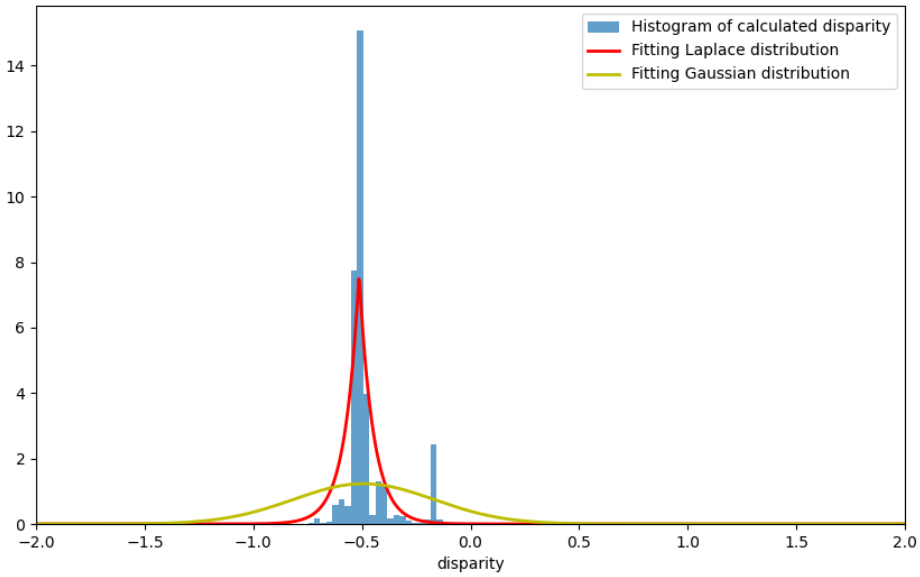


Figure 7. **Histogram of calculated disparity and fitting results of a probability distribution.** The histogram of disparity calculation results for samples where the ground truth disparity in DP is -0.5, and the fitting results of Gaussian and Laplace distributions to this histogram are shown. It can be seen that the Laplace distribution fits the error distribution better than the Gaussian distribution.

Figure 7 shows the histogram of disparity calculation results for 101,049 data samples where the ground truth disparity in DP is -0.5, along with the fitting results of Gaussian and Laplace distributions to this histogram. The error distribution forms a high-kurtosis histogram that sharply peaks near the ground truth, indicating that the Laplace distribution fits better than the Gaussian distribution. Additionally, the log-likelihood, which represents the goodness of fit between the distribution and the data, was -29,644 for the Gaussian distribution, whereas it was 104,774 for the Laplace distribution, confirming that the Laplace distribution is quantitatively a better fit. Therefore, to generate the error $n(\sigma_{\mathrm{d}})$, we sampled from the following zero-mean Laplace probability distribution function:

$$f(n;\sigma_{\mathrm{d}}) = \frac{1}{\sqrt{2}\sigma_{\mathrm{d}}} \exp\left(-\frac{\sqrt{2}|n|}{\sigma_{\mathrm{d}}}\right), \tag{1}$$

where n is the disparity error, and σ_{d} is the standard deviation of the error.

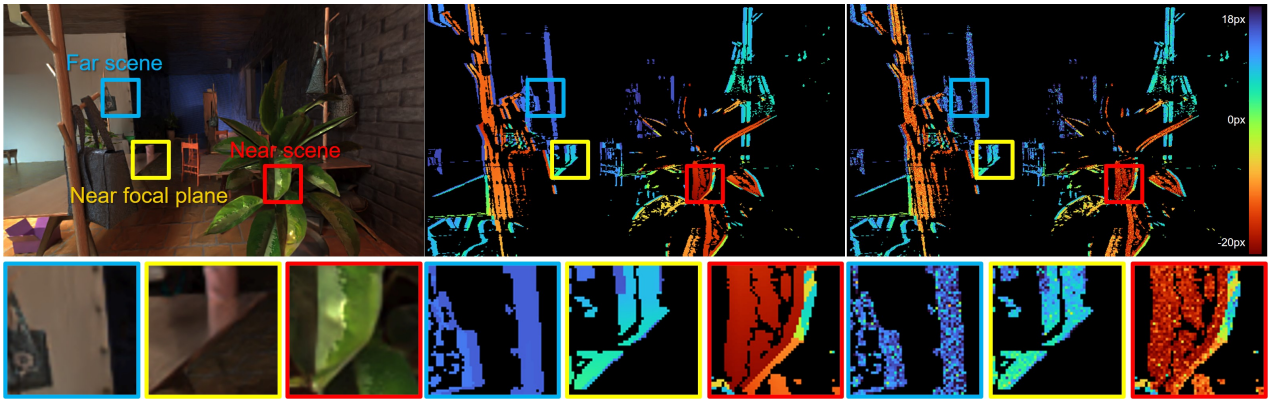
4.2. Details of disparity sampling

We discuss the details of the error sampling of disparity in Sec. 4.2 of the main study. We selected one training dataset from our synthetic dataset and show the disparity before and after error sampling in Fig. 8. The cyan frame shows the scene farther from the focal plane (far scene), the yellow frame shows the scene near the focal plane, and the red frame shows the scene in front of the focal plane (near scene). There are few errors near the focal plane, but many errors are added to the disparity in scenes far from the focal plane (near and far scenes). This trend is the same as the empirical error in Fig. 7 of the main study, suggesting that the error simulation works as expected.

An analytical perspective on disparity error: In Fig. 7 of the main study, we plot how the disparity error versus GT depth varies with focus distance. Furthermore, when the focus distance is fixed at 1.0 m, Fig. 9 (b) shows how the disparity error varies with the f-number (Fig. 9 (a) is a reprint of Fig. 7 of the main study).

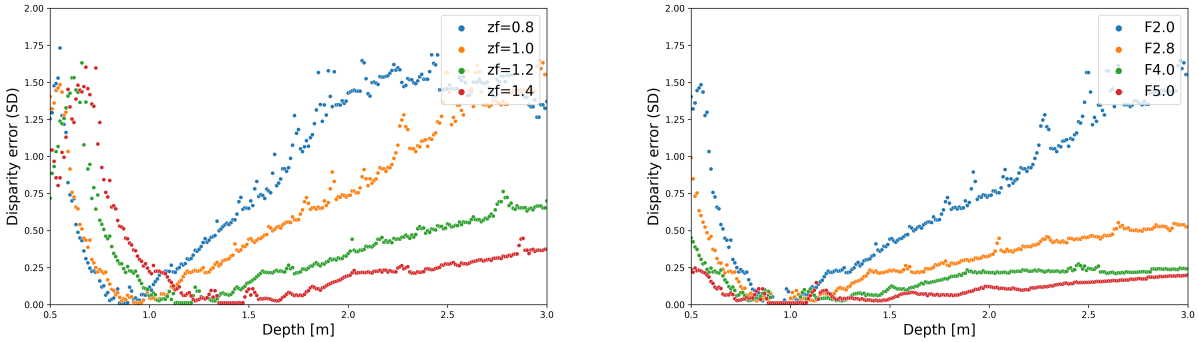
We have obtained the disparity error in this way under various conditions and have the following observations on its trend.

- The error is minimal when the subject is at a distance equivalent to the focal plane.



(a) RGB image (b) Disparity map (c) Disparity map after error sampling

Figure 8. **Disparity before and after error sampling.** The cyan frame shows the scene farther from the focal plane (far scene), the yellow frame shows the scene near the focal plane, and the red frame shows the scene in front of the focal plane (near scene). There are few errors near the focal plane, but many errors add to the disparity in scenes far from the focal plane (near and far scenes). This trend is the same as the empirical error in Fig. 7 of the main study, suggesting that the error simulation works as expected.



(a) TM error against depth for each focus distance (b) TM error against depth for each f-number

Figure 9. **TM error versus depth for each condition.** (a) Disparity error when the f-number is fixed at f/2.0 and the focus distance is varied. (b) Disparity error when the focus distance is fixed at 1.0 m, and the f-number is varied.

- The further the subject is from the focal plane, the larger the error becomes, but the error rises more slowly on the far side of the focal plane than on the near side.
- The larger the f-number, the smaller the disparity error.

The above observations, in conjunction with the disparity trend in Eqn. 1 of the main study, indicate that as disparity increases, the error increases, and as disparity decreases, the error decreases.

References

[1] Houdini. https://www.sidefx.com/products/houdini/. (Accessed on 07/17/2024). 3

[2] Unreal Engine Marketplace. https://www.unrealengine.com/marketplace/. (Accessed on 07/17/2024). 3

[3] 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

[4] Daniel Camozzato, Leandro Dihl, Ivan Silveira, Fernando Marson, and Soraia R Musse. Procedural floor plan generation from building sketches. The Visual Computer, 31(6):753–763, 2015. 3

[5] Jaewon Kam, Jungeon Kim, Soongjin Kim, Jaesik Park, and Seungyong Lee. Costdcnet: Cost volume based depth completion for a single rgb-d image. In European Conference on Computer Vision, pages 257–274. Springer, 2022. 2

[6] 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. 5

[7] 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 (CVPR), pages 4340–4349, June 2021. 5

[8] Jinsun Park, Kyungdon Joo, Zhe Hu, Chi-Kuei Liu, and In So Kweon. Non-local spatial propagation network for depth completion. In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16, pages 120–136. Springer, 2020. 3

[9] 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. 4, 5

[10] 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. 3

[11] 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. 5