DoF-Gaussian: Controllable Depth-of-Field for 3D Gaussian Splatting

Supplementary Material

To see the dynamic effect of our method and visual comparisons, please refer to our supplementary video. This document includes the following contents:

- Details of our synthetic dataset.
- Correctness of the proposed dataset.
- Color space
- Details on all-in-focus experiments.
- Details of ablation studies.
- Processing time.
- Limitations.

A. Details of our synthetic dataset

To quantitatively evaluate the refocusing ability and assess whether models learn accurate lens parameters, we introduce a synthetic dataset based on Real Forward Facing dataset [5] and Tanks and Temples dataset [2]. Specifically, we apply a state-of-the-art depth estimation method [1] to generate disparity maps from input images. Subsequently, we employ a single-image DoF rendering method [7], feeding both the input images and disparity maps into the network to produce images with bokeh blur, as shown in Fig. 1. We choose [7] to synthesize shallow DoF images primarily because it is predominantly based on traditional physical renderer despite the incorporation of neural networks. The rendered circle of confusion (CoC) in this approach will not be significantly differ from the CoC produced by our lens-based physical imaging model. In addition, we excluded Drjohnson and Playroom, two indoor 360° scenes, due to significant monocular depth estimation errors of multi-view input images in indoor environments. At the same time, the inability to generate *poses_bounds.npy* files for the Train and Truck scenes prevents the evaluation of DoF-NeRF on these two scenes. We maintain these two scenes for comparisons with future 3D-GS methods. To assess whether the model learns the exact aperture size $\mathcal{A}$ and focus distance$\mathcal{F}$ for each input image, we set these parameters artificially in advance. For the focus distance we set three cases, $\mathcal{F} = 0.2$, $\mathcal{F} = 0.5$ and $\mathcal{F} = 0.8$, corresponding to focus on the background, mid-ground, and foreground, respectively. Recognizing that the aperture size is closely related to the image resolution, we here normalize it to $0 - 1$ to facilitate the calculation of the error. We consider two cases for aperture size: $\mathcal{A} = 0.5$ and $\mathcal{A} = 1$. When we have optimized the 3D-GS scene, we get the learned focus distance and aperture size for each training image. Now, we can we can calculate

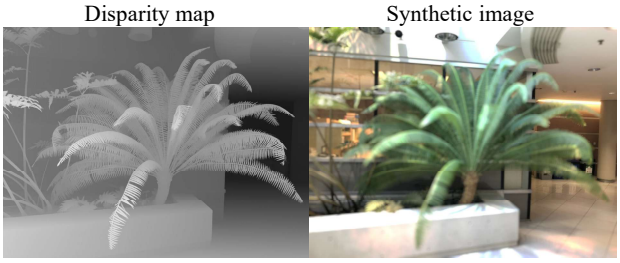

Figure 1. We show the disparity map generated by [1] and the synthetic shallow DoF image obtained from [7].

Table 1. Detailed comparison of our method and DoF-NeRF [8] on our synthetic dataset.

Method	DoF-NeRF [8]			Ours		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Fern	24.80	0.736	0.200	28.41	0.867	0.098
Flower	27.11	0.82	0.213	29.43	0.889	0.070
Fortress	29.78	0.846	0.186	32.22	0.926	0.059
Horns	24.23	0.812	0.235	27.64	0.863	0.122
Orchids	19.99	0.608	0.213	21.54	0.659	0.165
Room	26.55	0.842	0.198	32.16	0.933	0.071
Trex	26.65	0.853	0.207	29.53	0.910	0.082
Train	—	—	—	22.71	0.676	0.216
Truck	—	—	—	21.09	0.675	0.312

the lens parameter error as:

$$\delta_{\mathcal{A}} = \sum_{i}^{N} \frac{1}{N} |\mathcal{A}_i - \hat{\mathcal{A}}_i|, \tag{1}$$

where $\mathcal{A}$ and $\hat{\mathcal{A}}$ indicate the preset aperture size and learned aperture size, respectively, and N means the number of training images. The smaller this error $\delta_{\mathcal{A}}$ is, the more accurate our learned aperture size is. Similarly we use the following formula to calculate the focus distance error:

$$\delta_{\mathcal{F}} = \sum_{i}^{N} \frac{1}{N} |\mathcal{F}_i - \hat{\mathcal{F}}_i|, \tag{2}$$

where $\mathcal{F}$ and $\hat{\mathcal{F}}$ are the preset focus distance and learned focus distance. We use these two metrics to assess whether the model has learned the correct lens parameters. As demonstrated in Tables 1 and 2, our method outperforms DoF-NeRF [8] in both refocusing ability and the accurate estimation of lens parameters. Furthermore, as illustrated in Fig. 2, our method generates novel views that are more faithful to the ground-truth images.

Compared to the previous datasets proposed by Ma *et al*. [4] and Wu *et al*. [8], which evaluate defocus deblurring

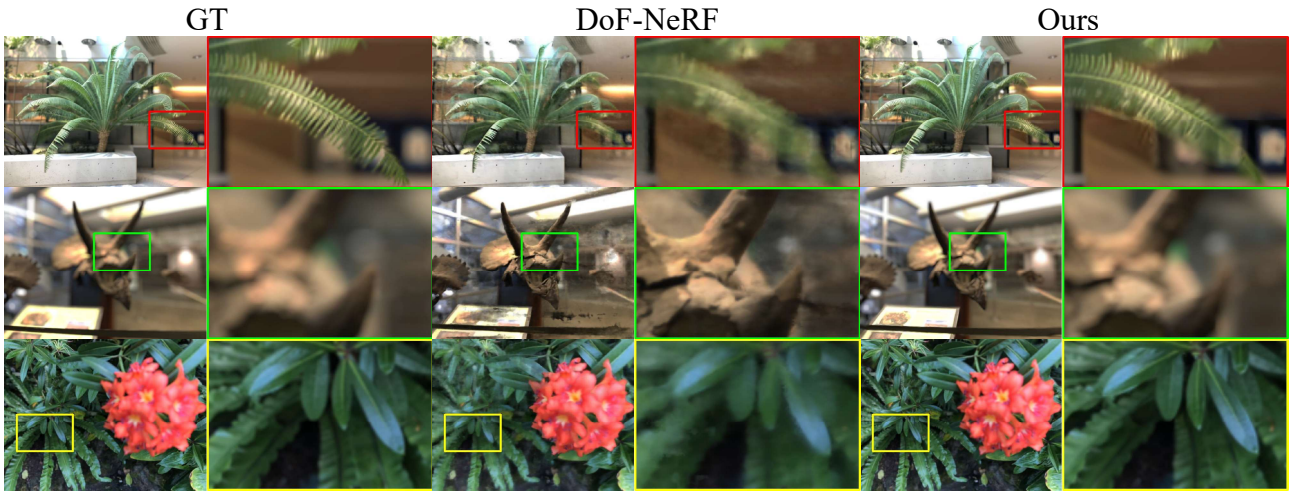

Figure 2. Visual comparison on our synthetic dataset. Our method performs significantly better than DoF-NeRF.

ability, our dataset is specifically designed to assess refocusing capabilities. Both the training and test sets in our synthetic dataset consist of shallow DoF images. We hope that this dataset will facilitate future work in this field.

Table 2. Detailed comparison of our method and DoF-NeRF [8] on our synthetic dataset.

Method	DoF-NeRF [8]		Ours	
	$\delta_{\mathcal{A}} \downarrow$	$\delta_{\mathcal{F}} \downarrow$	$\delta_{\mathcal{A}} \downarrow$	$\delta_{\mathcal{F}} \downarrow$
Fern	0.204	0.263	0.089	0.102
Flower	0.127	0.280	0.091	0.074
Fortress	0.156	0.299	0.187	0.021
Horns	0.234	0.205	0.197	0.075
Orchids	0.189	0.219	0.097	0.087
Room	0.276	0.278	0.066	0.116
Trex	0.189	0.251	0.154	0.079
Train	—	—	0.225	0.113
Truck	—	—	0.258	0.148

B. Correctness of the proposed dataset

We validate the accuracy of the synthesis strategy using the BLB dataset, which comprises 500 test samples, each containing paired all-in-focus and defocus images. All-in-focus images are processed through our synthesis pipeline, and the resulting synthesized defocus images are compared with the ground truth to calculate PSNR and SSIM metrics. The High PSNR and SSIM values indicate that the synthesized bokeh is close to the real, thereby confirming the effectiveness of our synthesis strategy.

C. Color space.

We apply a gamma transform on the input image to convert it from sRGB color space to linear color space. Subsequently, we simulate the circle-of-confusion within the linear color space. Finally, gamma correction is performed to convert the image from linear space back to sRGB space. The gamma value is 2.2. This process will be further emphasized in our revised version.

Table 3. The High PSNR and SSIM values indicate that our synthesized bokeh is close to the real.

	PSNR↑	SSIM↑
Ours	43.30	0.9932

D. Details on all-in-focus experiments

As shown in Table 4, we present the per-scene breakdown results of Real Forward-facing [5] and T&T_DB [2] datasets. These results align with the averaged results presented in the main text. Our method is built upon Mip-Splatting [9], a robust 3D-GS approach for all-in-focus inputs. Evidently, our method demonstrates superior performance compared to Mip-Splatting in most scenes. This indicates that our method can not only handle shallow DoF inputs, but also performs excellent under general input conditions, specially on Real Forward-facing dataset.

E. Details of ablation studies

In this section, we present detailed results of the ablation experiments in our main paper. In Table 5, we show the per-scene breakdown results of the ablation studies—baseline, w/o lens-based imaging model, w/o per-scene depth priors, w/o defocus-to-focus adaptation, sparse depth supervision, and no fine-tuned depth supervision. This indicates that each component of our system plays an important role in

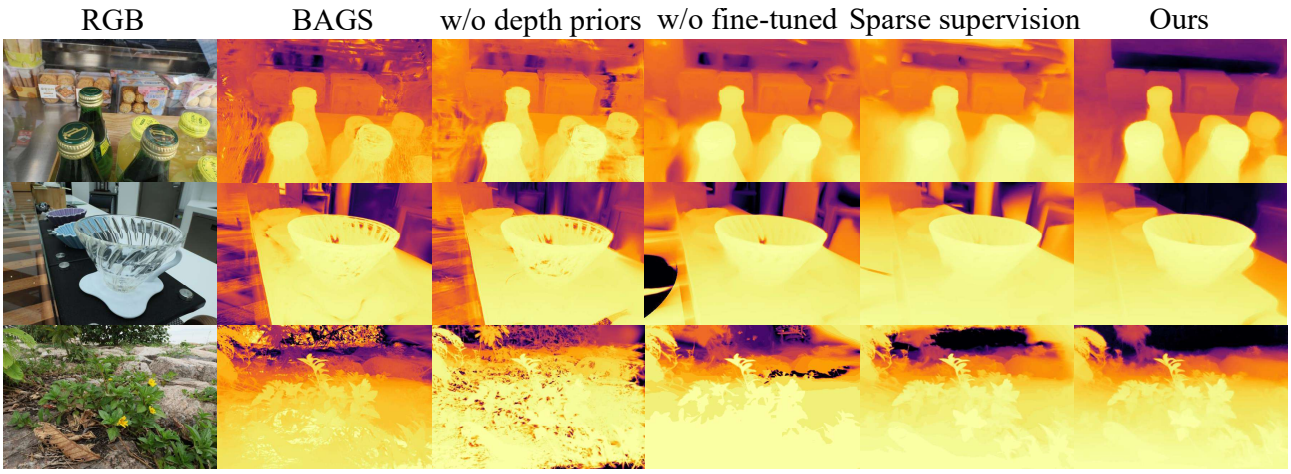

Figure 3. Visual comparison of different depth supervision strategies.

Table 4. Detailed comparison of other methods and ours on the all-in-focus dataset.

Method	Mip-Splatting [9]			Ours		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Fern	27.87	0.910	0.062	**28.23**	**0.917**	**0.062**
Flower	25.55	0.868	0.101	**27.81**	**0.910**	**0.059**
Fortress	27.85	0.913	0.072	**32.57**	**0.951**	**0.032**
Horns	29.47	0.952	**0.049**	**30.29**	**0.954**	0.052
Leaves	**22.42**	0.848	0.091	22.36	**0.850**	**0.089**
Orchids	21.58	0.800	0.105	**22.10**	**0.821**	**0.090**
Room	33.92	0.963	0.057	**34.63**	**0.973**	**0.052**
Trex	28.67	0.951	0.056	**30.29**	**0.961**	**0.042**
Train	**21.89**	**0.821**	**0.190**	21.15	0.791	0.251
Truck	**25.43**	**0.888**	**0.129**	24.61	0.872	0.180
Playroom	30.50	0.916	**0.223**	**30.93**	**0.924**	0.242
Drjohnson	**29.44**	0.890	**0.249**	29.37	**0.895**	0.258

improving the image deblurring quality. In addition, we demonstrate the effectiveness of our approach by showing a visual comparison of the depth maps rendered by 3D-GS under different depth strategies, as shown in Fig. 3.

F. Processing time.

As shown in Table 6, we recorded the processing time for both our method and other approaches on a single NVIDIA RTX A6000 GPU. For both Deblur-NeRF [4] and DoF-NeRF [8], we follow the specified training iterations outlined in the original papers, and calculate the training time. Due to the underlying NeRF-based framework, their average training time on the defocus deblurring dataset [4] is approximately 20 hours and 11 hours, respectively. Furthermore, their inference time is observed to be notably slow, achieving frame rates below 1 FPS. For the 3D-GS methods—BAGS [6], Deblurring 3DGS [3], and our method, we uniformly train for 30k iterations and record the training time and FPS. Although our method incorporates a lens imaging model, the training time is only slightly affected and it remains faster than BAGS. Benefit from the 3DGS framework, all GS-based methods can achieve fast rendering, obtaining FPS of approximately 360. In addition to training 3D Gaussian Splatting model, it takes about 3 minutes to fine-tune the depth network for per-scene depth priors.

G. Discussion on depth supervision.

We employ per-scene adjustments of depth priors to guide the reconstruction and ensure the accurate scene geometry and rendered depth maps. The effectiveness of this approach is demonstrated by ablation experiments. However, depth maps predicted by the fine-tuned depth network are not entirely accurate, and using these as pseudo-gt to supervise the depth maps rendered by 3D-GS introduces a degree of noise. This residual noise may impact the precision of the final depth maps, particularly in scenes with complex geometry. We therefore use a strategy of gradual decay of the depth loss weight w_d. In particular, we gradually decay this weight to $1/10$ of the initial value.

H. Limitations

Our method may encounter limitations when the blur is view-consistent, such as in cases where the camera maintains a fixed focal point, *i.e.*, focusing on a single target). Specifically, when the multi-view inputs all focus on the foreground, our method may struggle to recover clear background information. Consequently, a sharp scene can only be reconstructed if the input images contain both focused foreground and focused background elements. Addressing defocus deblurring under view-consistent conditions may be feasible through the integration of image priors, which we consider as a direction for future work.

Table 5. Ablation studies of per-scene breakdown results on the defocus deblurring dataset [4].

Method	baseline			w/o lens			w/o depth			w/o adaptation			sparse depth			w/o fine-tuned depth		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
cake	24.15	0.710	0.216	25.69	0.782	0.131	26.51	0.800	0.116	26.09	0.794	0.107	26.43	0.795	0.121	26.41	0.792	0.133
caps	21.24	0.559	0.332	23.57	0.713	0.148	24.41	0.741	0.145	24.12	0.737	0.142	24.52	0.742	0.164	24.52	0.745	0.157
cisco	20.77	0.732	0.114	20.85	0.743	0.069	20.95	0.742	0.071	20.88	0.739	0.067	20.72	0.734	0.079	20.76	0.736	0.082
coral	19.66	0.568	0.288	19.51	0.599	0.147	19.86	0.608	0.122	19.71	0.602	0.133	19.87	0.605	0.132	19.89	0.603	0.132
cupcake	21.72	0.686	0.198	22.09	0.742	0.089	22.82	0.757	0.079	22.63	0.752	0.080	22.74	0.752	0.0087	22.81	0.752	0.086
cups	24.29	0.749	0.223	25.89	0.814	0.100	25.91	0.818	0.114	26.06	0.820	0.086	25.34	0.800	0.115	25.63	0.804	0.117
daisy	18.00	0.493	0.299	23.35	0.734	0.062	23.33	0.721	0.086	23.54	0.724	0.069	22.88	0.706	0.114	22.80	0.700	0.119
sausage	17.45	0.461	0.284	17.99	0.515	0.169	18.47	0.536	0.151	18.29	0.529	0.156	18.18	0.531	0.172	18.55	0.550	0.153
seal	20.71	0.561	0.288	24.34	0.744	0.114	25.54	0.790	0.105	25.34	0.781	0.088	26.17	0.805	0.095	26.10	0.804	0.097
tools	25.09	0.845	0.152	27.17	0.898	0.056	28.09	0.911	0.051	27.53	0.902	0.052	27.57	0.900	0.061	27.82	0.902	0.059

Table 6. comparisons on processing time.

Method	Deblur-NeRF [4]	DoF-NeRF [8]	BAGS [6]	Deblurring 3DGS [3]	**Ours**
Time	20 hours	11 hours	25 mins	10 mins	18 mins
FPS	< 1	< 1	332	381	364

References

[1] Aleksei Bochkovskii, Amaël Delaunoy, Hugo Germain, Marcel Santos, Yichao Zhou, Stephan R Richter, and Vladlen Koltun. Depth pro: Sharp monocular metric depth in less than a second. *arXiv preprint arXiv:2410.02073*, 2024. 1

[2] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM TOG*, 42(4), 2023. 1, 2

[3] Byeonghyeon Lee, Howoong Lee, Xiangyu Sun, Usman Ali, and Eunbyung Park. Deblurring 3D Gaussian Splatting. *arXiv preprint arXiv:2401.00834*, 2024. 3, 4

[4] Li Ma, Xiaoyu Li, Jing Liao, Qi Zhang, Xuan Wang, Jue Wang, and Pedro V Sander. Deblur-NeRF: Neural radiance fields from blurry images. In *CVPR*, 2022. 1, 3, 4

[5] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1), 2021. 1, 2

[6] Cheng Peng, Yutao Tang, Yifan Zhou, Nengyu Wang, Xijun Liu, Deming Li, and Rama Chellappa. BAGS: Blur Agnostic Gaussian Splatting through Multi-Scale Kernel Modeling. *arXiv preprint arXiv:2403.04926*, 2024. 3, 4

[7] Juewen Peng, Zhiguo Cao, Xianrui Luo, Hao Lu, Ke Xian, and Jianming Zhang. Bokehme: When neural rendering meets classical rendering. In *CVPR*, 2022. 1

[8] Zijin Wu, Xingyi Li, Juewen Peng, Hao Lu, Zhiguo Cao, and Weicai Zhong. DoF-NeRF: Depth-of-field meets neural radiance fields. In *ACM MM*, 2022. 1, 2, 3, 4

[9] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-splatting: Alias-free 3D gaussian splatting. In *CVPR*, 2024. 2, 3