Reconstructing Nerf with Depth-Of-Field Images

Qi Wen Gan
Department of CST
Tsinghua University
2021280382

Yu-Wen Michael Zhang
Department of CST
Tsinghua University
2022280391

Ruiqi Zhang
Department of Mathematical Sciences
Tsinghua University
2022311353

1 Introduction

Recently, Neural Field emerges as a huge interdisciplinary research field that encapsulates the concept of 3D vision, computer graphics, and neural networks. Topics such as 3D Generation, 3D Reconstruction, Nerf, and 3D Human are commonly seen in Neural Field. Our proposed topic, "Reconstructing Nerf with different Depth-Of-Field Images" focuses on implicitly reconstructing 3D scenes given a sparse input of 2D images that have different focal lengths and depths of field (See Appendix A Figure 1). The main objective of this task is to find a reliable algorithm that is capable of dealing with data sparsity, different focal lengths and depth-of-field, and noisy sensory data. Ultimately, this task will help discover the potential of positional embeddings as it can be an extremely good fit for the needs of fitting 3D coordinate's features and also the wider domain of image vision tasks.

Neural radiance fields (NeRF) encode a scene that requires a large number of input views and use tone-mapped low dynamic range (LDR) as input. In order to overcome those defects, Roessle et al. propose a method that guides the NeRF optimization with dense depth priors, without the need for additional depth input. Their method enables novel view synthesis with NeRF on room-size scenes using only 18–36 images. Mildenhall et al.(5) modify NeRF to instead train directly on linear raw images, preserving the scene's full dynamic range. By rendering raw output images from the resulting NeRF, they can perform novel high dynamic range (HDR) view synthesis tasks.

In this report, we propose a new NeRF coordinate embedding method using a deep learning method, specifically using Latent Fourier Feature method. Our proposed method is capable to learn and treat the shallow Depth of Field (DoF) images as high-frequency details, thus, enhancing the visual quality of the 3D neural rendering and scene reconstruction in a Shallow DoF setting.

To demonstrate comprehensive experimental results, we use the popular 3D scene reconstruction datasets, namely, LLFF dataset (6) in our study. We also evaluate and compare the performance of our method with DoF-NeRF (11) works. We will release our code and tensor board results at https://github.com/qiwen98/NeRF-DoF.

Our contributions are summarized as follows:

1. We propose a new NeRF coordinate embedding method using deep learning method, specifically using Latent Fourier Feature method.
2. We show the qualitative geometry reconstruction details, e.g. depth of the scene is superior to our baseline work under both All-in-Focus and Shallow DoF settings.
3. We perform a comprehensive quantitative and visualized qualitative analysis on the reconstructed NeRF to show that our proposed model outperforms our baseline.

The rest of this article is organized as follows. Section 2 reviews some previous works related to reconstructing NeRF from images with different camera parameters and coordinate embedding methods. Section 3 introduces the procedure that we used in this study and some preliminary knowledge in NeRF representation. Section 4 details the experiments and results, followed by the conclusion in Section 5.

Preprint. Under review.

2 Related Work

Reconstuct NeRF from images with different camera parameters: Another variant is called Deblur-NeRF, which is used to recover a sharp NeRF from blurry pictures. Ma et al (3) found that when using a long exposure setting to capture a low-light scene, the images are more sensitive to camera shake, resulting in camera motion blur. Moreover, when capturing scenes with large depth variation using a large aperture, defocus blur is inevitable. In order to address this problem, Ma et al. propose Deblur-NeRF, an effective framework that explicitly models the blurring process in the network and is capable of restoring a sharp NeRF from blurry input.

The second common problem of NeRF and its variants are generally assumed all-in-focus inputs that are generated based on the pinhole camera model, the model performs poorly when processing shallow DoF and out-of-focus input images. Therefore, Wu et al (11)introduce a new model called Depth-of-Field Neural Radiance Field (DoF-NeRF) in order to address the problem where pictures have finite depth-of-field. DoF-NeRF is a novel neural rendering approach that can deal with shallow DoF inputs and can simulate the effect of DoF. It extends NeRF to simulate the aperture of the lens following the principle of geometric optics. Moreover, it enables direct manipulation of the DoF effect by adjusting the virtual aperture and focus parameters. In this proposed work, the author jointly optimize the aperture parameters and focus distances parameters to produce a smooth and natural DoF "bokeh" effect. However, this work doesn't account for the normalized depth of the scene when optimizing the parameters. Besides, the author acknowledges that although the disparity map can be predicted by depth estimation, its accuracy is not guaranteed. We hypothesize that manipulating the coordinate embedding methods can produce higher fidelity of NeRF reconstruction in different aperture and focus distance settings.

Coordinate Embedding and Fourier Feature (FF):

Fourier feature(FF) was first proposed in the seminal work of Rahimi and Recht (10), which constructs an explicit feature map such that the inner product of the explicit feature vectors can be used to approximate the kernel function. On the basis of their work, we use Fourier feature mapping which is an input mapping that transforms input coordinates before passing them into the following model. Specifically, this mapping input points in d dimension Hypercube to the surface of a higher dimensional hypersphere with a series of sine and cosine signals, which allows networks to process high-frequency functions when dealing with low-dimensional problem domains and can improve the performance of our model. These features are typically used in computer vision and graphics tasks where accurate representations of complex 3D objects and scenes are needed. On the basis of Fourier Features, we extract latent variables such as those computed from observed features using matrix factorization, then we get the Latent Fourier Feature (LFF). The latent Feature was first introduced in the paper of styleGAN (2). Ivan et al. (1) further enhanced the method of Latent feature given the value of a random latent vector and the coordinate of that pixel, we hypothesise that using the same method, we can incorporate the latent space into the standard NeRF coordinate embedding procedure, with the name of Latent Fourier Feature (LFF). We present our method in 3 and we show our results in 4

3 Method

Volumetric Radiance Fields: Volumetric radiance fields typically consist of density and radiance. The density field maps the spatial position $x \in \mathbb{R}^3$ to the opacity $\sigma(x) \in \mathbb{R}$ at a point. The radiance field L defines the RGB colour $L(x, d) \in [0, 1]^3$ emitted at point x in view direction $d \in \mathbb{R}^3$. In modern NeRF works, these fields are usually represented using a multilayer perceptron (MLP)(7). Others traditional sparse spatial data structures, especially inside an associated bounding box and voxel liked grid space, such as octrees (13; 12), have also been explored. In NeRF (7), an arbitrary camera ray is calculated with the equation of $r(t) = x + td$, these rays is then used compute the color $c(r)[0, 1]^3$ using the volume-rendering equation as

$$\mathbf{c}(\mathbf{r}) = \int_0^\infty T(t; \mathbf{r})\sigma(\mathbf{r}(t))L(\mathbf{r}(t), \mathbf{d})\mathrm{d}t \tag{1}$$

where $T(t; r) \in \mathbb{R}$ denotes the transmittance function

$$T(t;\mathbf{r}) = \exp\left(-\int_0^t \sigma(\mathbf{r}(s))\mathrm{d}s\right) \tag{2}$$

The integral in 1 is numerically computed using the quadrature rule for volume rendering (4). Specifically, a finite number of points $t_0 = 0 < t1 < \ < t_N \in \mathbb{R}$ are first sampled along the ray and the approximated color $\hat{c}(r)[0,1]^3$ is computed as a Riemann sum

$$\hat{\mathbf{c}}(\mathbf{r}) = \sum_{i=1}^{N} T_i \left(1 - \exp\left(-\sigma_i \delta_i\right)\right) L_i, \tag{3}$$

$$T_i = \exp\left(-\sum_{j=0}^{i-1} \sigma_j \delta_j\right) \tag{4}$$

$\sigma_i = \sigma(r(s_i))$ and $L_i = L(r(s_i))$ for some $s_i \in [t_i, t_i + 1]$, where $\delta i = t_{i+1} t_i$ are the lengths of the corresponding intervals. The accumulated transmittance T_N is used to composite $\hat{c}(r)$ and the background, that is, the final colour is obtained as $(1 - T_N)\hat{c}(r) + T_N c0$, where c_0 is the background colour typically given as a hyperparameter.

Positional Embedding: In recent work, the use of positional embedding in image synthesis tasks has been prevalent. The work of **Image Generators with Conditionally-Independent Pixel Synthesis** (1)demonstrated very good results in image synthesis given the value of a random latent vector and the coordinate of that pixel. Presumably, this idea can be extended to our proposed work in 3D novel view synthesis using NeRF. In Anokhin's work, the combination of Fourier embedding and coordinate embedding (H x W) learnable vector achieved high fidelity in image generation tasks. Our main contribution to this work is to improve the original positional embedding method introduced in the original NeRF Paper (10) 5.

$$\begin{aligned}\gamma(x,y) = [&\sin(\pi x), \cos(\pi x), \sin(\pi y), \cos(\pi y), \sin(2\pi x), \cos(2\pi x), \sin(2\pi y), \cos(2\pi y), \\ &\ldots\ldots, \sin\left(2^{L-1}\pi x\right), \cos\left(2^{L-1}\pi x\right), \sin\left(2^{L-1}\pi y\right), \cos\left(2^{L-1}\pi y\right)\end{aligned} \tag{5}$$

we use a Fourier Feature mapping 5to feature input coordinates before passing them through a coordinate-based MLP and investigate the theoretical and practical effect this has on convergence speed and generalization. The function maps input points $v[0,1)^d$ to the surface of a higher dimensional hypersphere with a set of sinusoidal functions. We then introduce a single-layer conv1d operation to increase the performance, the weight and biases in convd1d first map the 1 dim frequency input into 64 dim frequency with higher dimensional space. We apply the convolution with the hyperparameter of kernel size=1 and padding=0. We then apply a sinusoidal activation function onto it to aggregate the learned features back to 1 dim. The whole operation can be summarised in equation 6

$$out(N_i, C_{outj}) = bias(C_{outj}) + \sum_{k=0}^{c_{in}-1} weight(c_{out}, k) * input(N_i, k) \tag{6}$$

$$out(N_i, C_{outj}) = sin\left[Out(N_i, C_{outj})\right] \tag{7}$$

where * is the valid cross-correlation operator, N is the batch size, C denotes the number of channels, and L is the length of the signal sequence.

FineTuning: The use of conventional Fourier Feature coordinate embedding in Nerf will always preserve the high-frequency details from the image, however, given the out-of-focus, different field of view images, the blurred portion will be treated as high-frequency details, and it will degrade the visual quality of the 3D neural rendering and scene reconstruction. Therefore, We modified the Fourier feature coordinate embedding based on some random latent vector and the coordinate of that pixel using the method mentioned above. we initialize the Adam Optimizer with a 0.0005 learning rate. We used a learning scheduler and decay the learning rate in the 10 and 20 steps with a

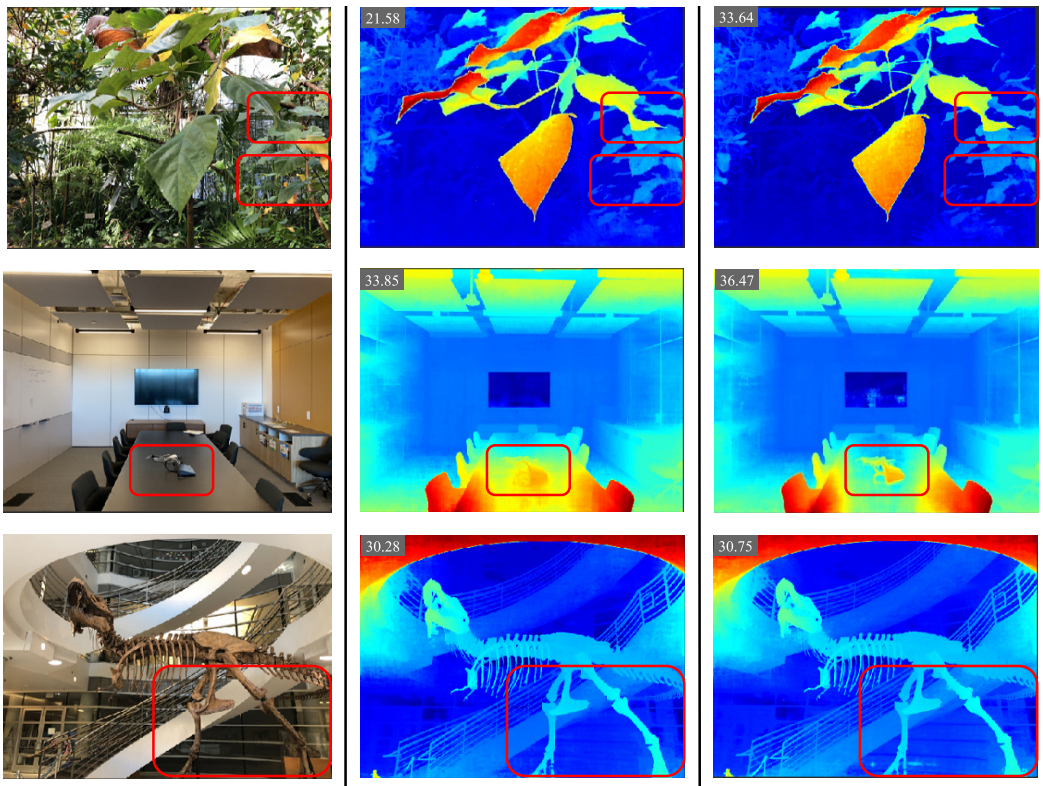

Figure 1: Example of using SVG on Overleaf

decay gamma of 0.5. In the original DoF-NeRF paper (11), the author found that the NeRF network was unable to reconstruct accurate geometry when training with a shallow Depth of Field Dataset, therefore a two-stage training method is required. Inspired by this quote, we first run the pre-training process with the MLP network using an All-in-focus data-set, by batching the ray in with a size of 1024. In every scene, the MLP will converge after approximately 3M iterations. After that, we then load the pre-trained weights and fine-tune the network with a shallow Depth of Field (DoF) data set with another 3M iteration using the same hyper-parameter that we used during the pre-training stage.

4 Experiment

Baseline: The top result from the DoF-NeRF had a PSNR of 30.33 (third column of Table 1 in Appendix A) for the real-world dataset and 26.10 (third column of Table 2 in Appendix A) for the synthetic dataset. The authors identified that rendering DoF effects from a single all-in-focus image has been well studied in previous work, but all of the works assume all-in-focus inputs. We shall continue to examine the challenges faced by the author in Dof-Nerf, allowing us to improve the PSNR, SSIM, and LPIPS of our proposed method.

Metrics: In our evaluation, we will compare the results from both the given baseline method and our proposed method. The evaluation method can be divided into two parts, qualitative and quantitative of NeRF reconstruction, and DoF Rendering, we present quantitative results of NeRF reconstruction with 4 metrics. The formulas of these metrics are all located in Appendix A.

Dataset: Our main Local Light Field Fusion (LLFF) dataset (6) are mainly taken from the Real Forward-Facing dataset (7) and it used as inputs and adopt the same initialization, training, and rendering settings as the experiments conducted on the shallow DoF data. Because of time and resources limit, we only include scenes of Trex, Room, Leaves, and Fortress. For the "Synthetic (Shallow Depth of Field)" dataset, we followed the data creation procedure from DoF-NeRF (11), which first created the disparity map using DPT (9), and then generate the shallow Depth of Field (DoF) image datasets using BokehMe (8).

Table 1: All in Focus Experiments Result

Scene	Model	PSNR ↑	SSIM ↑	LPIPS ↓
fortress	Ours LFF	**33.05**	**0.99**	**0.028**
	NeRF ✱(11)	33.97	0.94	0.046
	DOF-NeRF ✱ (11)	34.00	0.94	0.045
leaves	Ours LFF	**33.64**	**0.92**	**0.061**
	NeRF ✱ (11)	21.58	0.78	0.162
	DOF-NeRF ✱(11)	21.48	0.78	0.159
room	Ours LFF	**36.47**	**0.99**	**0.029**
	NeRF ✱ (11)	33.85	0.95	0.072
	DOF-NeRF ✱(11)	33.76	0.95	0.073
trex	Ours LFF	**30.75**	**0.97**	**0.061**
	NeRF ✱(11)	30.28	0.95	0.060
	DOF-NeRF ✱ (11)	30.22	0.94	0.060

✱ results taken directly from DOF-NeRF paper

Table 2: "Synthetic (**Shallow Depth of Field**)" Experiments Result

Scene	Model	PSNR ↑	SSIM ↑	LPIPS ↓
fortress	Ours LFF	**34.30**	**0.92**	**0.061**
	Ours finetune w/Relu	33.90	0.90	0.064
	NeRF ✱ (11)	28.14	0.78	0.201
	DOF-NeRF ✱(11)	29.16	0.81	0.183
leaves	Ours LFF	**25.10**	**0.82**	**0.186**
	Ours finetune w/Relu	25.08	0.81	0.189
	NeRF ✱(11)	19.45	0.65	0.319
	DOF-NeRF ✱ (11)	20.02	0.70	0.276
room	Ours LFF	30.95	**0.95**	0.159
	Ours finutune w/Relu	**31.72**	0.94	0.188
	NeRF ✱(11)	26.66	0.87	0.196
	DOF-NeRF ✱(11)	29.44	0.91	**0.150**
trex	Ours LFF	**30.49**	**0.93**	**0.144**
	Ours finetune w/Relu	25.31	0.85	0.261
	NeRF ✱(11)	24.43	0.83	0.172
	DOF-NeRF ✱ (11)	25.72	0.87	0.156

✱ results taken directly from DOF-NeRF paper

Improvement in All-In-Focus setting: Here we present the comparison on the all-in-focus dataset. To additionally validate the effectiveness of our method in the all-infocus setting, we design an experiment where wide DoF images from the Real Forward-Facing dataset (7) are used as inputs and adopt the same initialization, training, and rendering settings as the experiments conducted on the shallow DoF data. Table 1 reports the quantitative comparison on the all-in-focus dataset with (7) and (11) results. Although our proposed Latent Fourier Feature (LFF) is originally for shallow DoF inputs, experiments indicate that our method shows superior performance against and DoF method vanilla NeRF using all-in-focus images as inputs. In Figure 1, we can clearly observe that in the scene, leaves, room, and trex, by adding our proposed LFF, we can achieve finer geometry details in the background. Here we only compared the quantitative results taken directly from DOF-NeRF paper and the author did not realise their code. Our quantitative PSNR, SSIM, LPIPS also show superior results compared to the (7) and (11) results.

Improvement in Shallow Depth of Field setting: In this section, we compare our method qualitatively and quantitatively with DoF-NeRF on Synthetic (Shallow Depth of Field) datasets. In the original DoF-NeRF paper (11), the author found that the NeRF network was unable to reconstruct

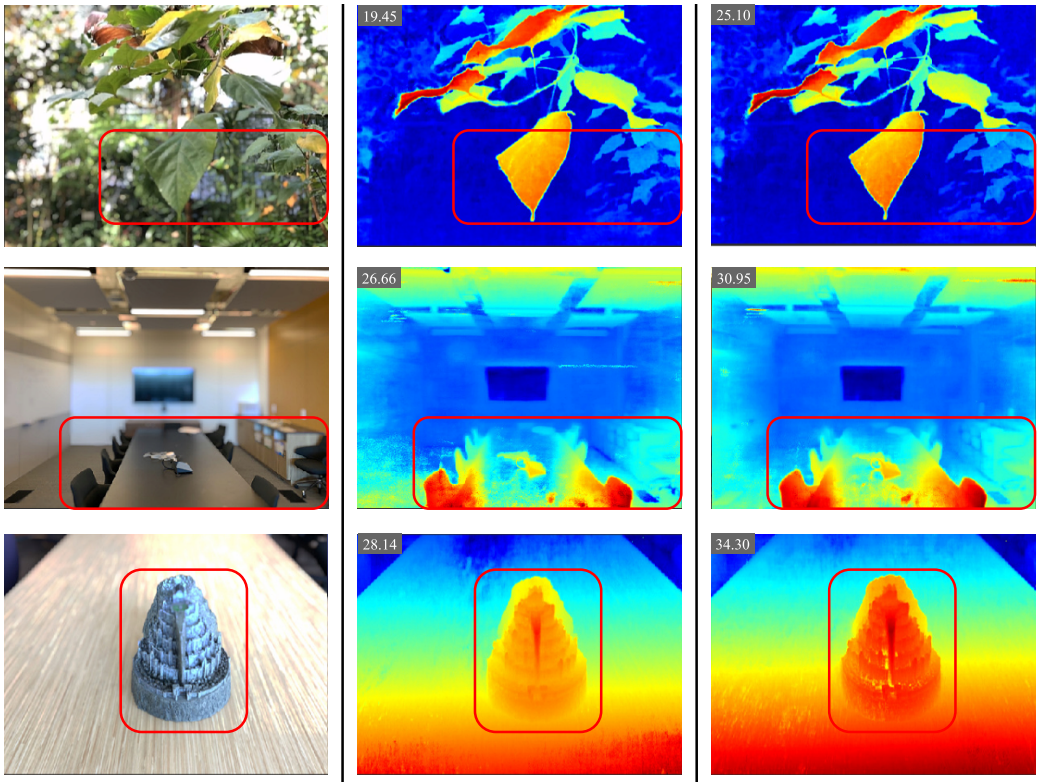

Figure 2: Example of using SVG on Overleaf

accurate geometry when training with a shallow Depth of Field Dataset, therefore a two-stage training method is required. In our experiment, we proved that, by simply fine-tuning the pre-trained model with the Relu activation function, we can get an additional quality boost in novel Shallow DoF view image reconstruction even without using the method proposed by (11). As shown in Table 2, one can observe that our method demonstrates better perceptual geometry qualities than DoF NeRF when rendering shallow DoF novel views on LLFF datasets. After applying the proposed LFF, we get significant improvement in the novel scene reconstruction, especially for the in-focus object part. In figure 2, we can see the geometry of the in-focus object achieving higher fidelity. This implies that our proposed Latent Fourier Feature enables NeRF to tackle shallow DoF inputs and benefits all-in-focus novel view synthesis. We wanted to include the qualitative comparisons with DoF-NeRF (11) in Figure 2 but the authors haven't realised their code. In future, we would definitely wish to reproduce the results of DoF-NeRF and compared them with our results.

5 Conclusion

In our experiments, we proposed to add a Latent Fourier Feature embedding on the top of the original proposed positional encoding method to enhance the Perceptual quality of the NeRF novel synthesis task. The proposed Latent Fourier Feature function act as a regularizer for capturing the high-frequency details in 3D shallow Depth of Field neural rendering and scene reconstruction. The Latent Fourier Feature is a simple function and also be extended into other concurrent NeRF works. Our experiment results show that we can achieve a significant improvement in both qualitative and quantitative results when we encode the viewing direction and the world space coordinate with the proposed new LFF function.

Contribution Attribution

Qi Wen Gan completed most of the implementation and execution of the experiment. He also drafted the Method, Experiments, and Conclusion part of this report.

Yu-Wen Michael Zhang completed and designed most of the poster. He also checked the grammar issue of this report and post-process the diagram provided by Qi Wen to make the report clean and readable.

Ruiqi Zhang supports the theoretical part of the project, including writing conceptual and derivations at the theoretical level.

References

[1] Ivan Anokhin, Kirill Demochkin, Taras Khakhulin, Gleb Sterkin, Victor Lempitsky, and Denis Korzhenkov. Image generators with conditionally-independent pixel synthesis. *CoRR*, abs/2011.13775, 2020.

[2] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *CoRR*, abs/1812.04948, 2018.

[3] Li Ma, Xiaoyu Li, Jing Liao, Qi Zhang, Xuan Wang, Jue Wang, and Pedro V. Sander. Deblur-nerf: Neural radiance fields from blurry images, 2021.

[4] N. Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108, 1995.

[5] Ben Mildenhall, Peter Hedman, Ricardo Martin-Brualla, Pratul Srinivasan, and Jonathan T. Barron. Nerf in the dark: High dynamic range view synthesis from noisy raw images, 2021.

[6] Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, 2019.

[7] 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. In *ECCV*, 2020.

[8] Juewen Peng, Zhiguo Cao, Xianrui Luo, Hao Lu, Ke Xian, and Jianming Zhang. Bokehme: When neural rendering meets classical rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[9] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020.

[10] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *CoRR*, abs/2006.10739, 2020.

[11] Zijin Wu, Xingyi Li, Juewen Peng, Hao Lu, Zhiguo Cao, and Weicai Zhong. DoF-NeRF: Depth-of-field meets neural radiance fields. In *Proceedings of the 30th ACM International Conference on Multimedia*. ACM, oct 2022.

[12] Alex Yu, Sara Fridovich-Keil, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. *CoRR*, abs/2112.05131, 2021.

[13] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. *CoRR*, abs/2103.14024, 2021.

A Appendix

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2$$

$$\begin{aligned} PSNR &= 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right) \\ &= 20 \cdot \log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right) \\ &= 20 \cdot \log_{10}\left(MAX_I\right) - 10 \cdot \log_{10}(MSE) \end{aligned}$$

$$\mathrm{SSIM}(x,y) = \frac{(2\mu_x\mu_y + c_1)\,(2\sigma_{xy} + c_2)}{\left(\mu_x^2 + \mu_y^2 + c_1\right)\left(\sigma_x^2 + \sigma_y^2 + c_2\right)}$$

$$LIPIS = \mathrm{d}(x, x_0) = \sum_l \frac{1}{H_l W_l}\sum_{h,w} ||w_l \odot (\hat{y}^l_{h,w} - \hat{y}^l_{0hw})||_2^2$$

The statistics of the experimental synthetic dataset's results taken from original DoF-NeRF paper(11) are as follows:

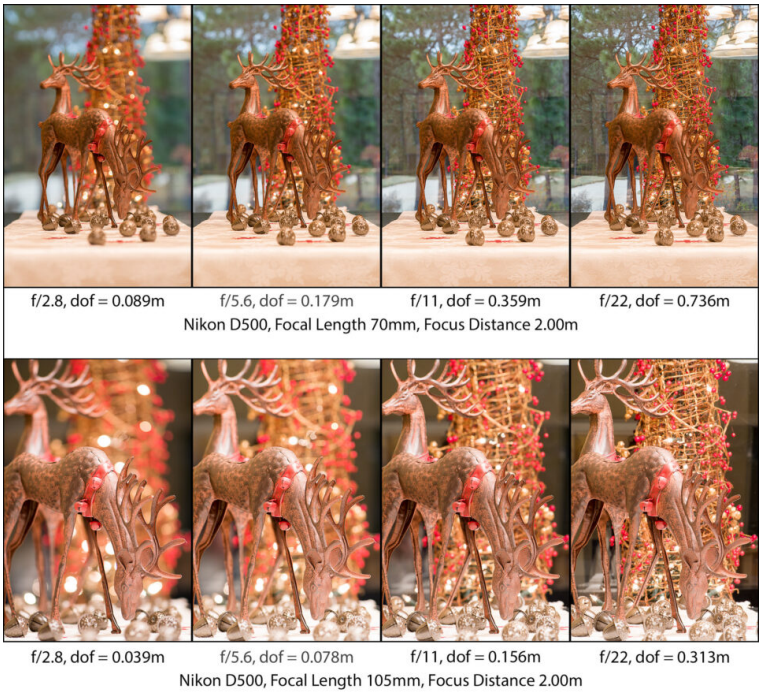

Figure 3: Images with different Depth of Field. Image Taken from `https://photographylife.com/what-is-depth-of-field`

Table 1: Comparison of NeRF [27] and our framework in the real-world dataset.

Scene	Model	PSNR(↑)	SSIM(↑)	LPIPS(↓)
amiya	NeRF	26.924	0.9092	0.1633
	ours	**28.311**	**0.9289**	**0.1370**
camera	NeRF	25.593	0.8862	0.1574
	ours	**27.714**	**0.9134**	**0.1259**
plant	NeRF	28.272	0.8961	0.1581
	ours	**30.317**	**0.9290**	**0.1178**
turtle	NeRF	33.531	0.9566	0.0939
	ours	**34.965**	**0.9647**	**0.0823**

Table 2: Comparison of NeRF [27] and our framework in the synthetic dataset.

Scene	Model	PSNR (↑)	SSIM(↑)	LPIPS(↓)
fortress	NeRF	28.142	0.7826	0.2011
	ours	**29.168**	**0.8099**	**0.1830**
leaves	NeRF	19.450	0.6541	0.3190
	ours	**20.025**	**0.7000**	**0.2766**
room	NeRF	26.668	0.8743	0.1961
	ours	**29.443**	**0.9135**	**0.1502**
trex	NeRF	24.433	0.8379	0.1723
	ours	**25.726**	**0.8744**	**0.1564**