SCIENCE CHINA
Information Sciences

• **Supplementary File** •

PD-NeRF : A General Pseudo-Depth Supervision Method for Neural Radiance Fields

Jiaming GU[1,2], Minchao JIANG[1], Xiaoyuan LU[3], Cong HUA[1], Hongsheng LI[1], Guangming ZHU[1] & Liang ZHANG[1*]

[1]*School of Computer Science and Technology, Xidian University, Xian* 710071*, China;*
[2]*QINGYI(Shanghai) ITS, Shanghai* 201306*, China;*
[3]*Shanghai Lingang SCDC, Shanghai* 201306*, China*

Appendix A Distinctions from DS-NeRF

In the paper, we mentioned: "Different from DS-NeRF [1], our method ensures a sufficient number of feature points, making it more suitable for high-resolution images without the need for resolution reduction during training. Moreover, our approach to depth estimation is more concise and efficient compared to DS-NeRF. Additionally, our NeRF model significantly reduces both training and inference times." **In this section, we will provide a detailed explanation of the differences between our method and DS-NeRF.**

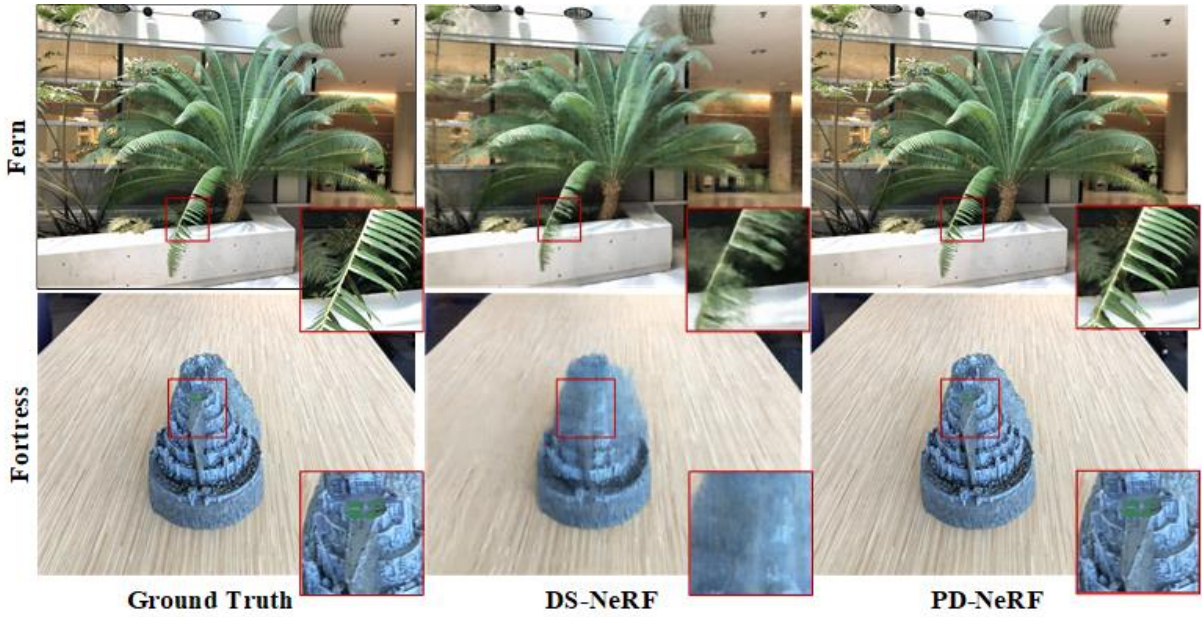

Figure A1 Qualitative Comparison between DS-NeRF and PD-NeRF. We visualize the rendering results of DS-NeRF and PD-NeRF on the LLFF dataset. PD-NeRF demonstrates noticeably superior performance compared to DS-NeRF, both in terms of fine details and overall quality.

DS-NeRF [1] proposes a method to provide depth supervision for NeRF by using point clouds obtained from COLMAP or RGB-D cameras. It employs the KL divergence to align the weight distribution along a ray, obtained from the Gaussian distribution of the depth, with the weight distribution derived from the NeRF volumetric rendering process. **The key differences between DS-NeRF and PD-NeRF are as follows:**

• The motivations behind the two methods differ. DS-NeRF proposes a scheme for depth supervision using point clouds, regardless of their source, while PD-NeRF aims to leverage point clouds generated during the COLMAP process for faster and more effective reconstruction and addresses the weak supervision issue associated with sparse point clouds.

• Although both DS-NeRF and PD-NeRF utilize Gaussian distributions, the purposes differ significantly. In DS-NeRF, the Gaussian distribution is used to model the weight distribution along a ray, whereas in PD-NeRF, it is used to estimate pseudo-depth for pixels without corresponding point clouds.

* Corresponding author (email: liangzhang@xidian.edu.cn)

• The source and processing of the point clouds differ. Since the point clouds obtained through COLMAP are sparse, especially for high-resolution images where the supervision is weak, DS-NeRF reduces the image resolution or leverages dense point clouds provided by RGB-D cameras for depth supervision. PD-NeRF addresses the issue of weak supervision from sparse point clouds by proposing a method to construct pseudo-depth, eliminating the need to downsample the original image resolution or use additional equipment to capture dense point clouds.

• The supervision strategies are different. DS-NeRF essentially supervises the weight distribution along a ray. The depth supervision used in DS-NeRF is based on KL divergence. It defaults the depth on each ray to satisfy theimpulse function (single-peaked normal distribution). As a result, DS-NeRF struggles with objects in 360-degree scenes that have 'double surfaces' (where a ray enters and exits the surface), limiting it to forward-facing datasets. In contrast, the supervision strategy in PD-NeRF can be applied to arbitrary scenes.

Table A1 Quantitative Comparison of DS-NeRF and PD-NeRF.PD-NeRF outperforms DS-NeRF across all evaluation metrics.

Methods	PSNR↑	SSIM↑	LPIPS↓	Training Time	FPS
DS-NeRF	24.9	0.72	0.34	∼5h	-
PD-NeRF	**26.5**	**0.90**	**0.21**	**∼10mins**	**∼15**

In Figure A1, we present a qualitative comparison of the rendering results of DS-NeRF and PD-NeRF on the LLFF dataset [2]. Since DS-NeRF is only compatible with forward-facing scenes, we trained PD-NeRF in forward-facing scenes, controlling for the same number of iterations (50,000). Compared to DS-NeRF, PD-NeRF achieves higher rendering quality, effectively capturing details such as the leaves in the Fern scene. In the Fortress scene, DS-NeRF's rendered results exhibit noticeable blurriness, while PD-NeRF demonstrates high fidelity compared to the Ground Truth. Table A1 presents a quantitative comparison between DS-NeRF and PD-NeRF. PD-NeRF outperforms DS-NeRF across all metrics. Notably, PD-NeRF converges significantly faster than DS-NeRF in terms of training time. Furthermore, DS-NeRF does not support real-time rendering, while PD-NeRF achieves a real-time rendering speed of 15 FPS.

Appendix B Ablation Study

We will compare the impact of feature points across different pixel coverage ranges on the results. In the paper, we mentioned: "Where f is the scaling factor, ensuring that the pixels in the area covered by all feature points occupy about 6% of the entire image. It has been verified that the sampling points occupying 6% of the entire picture can achieve a good supervision effect and the overhead is very small. When the image pixels are $1600 * 1600$, $f = 1$." We will now further elaborate and adjust the sampling point ratio for validation.

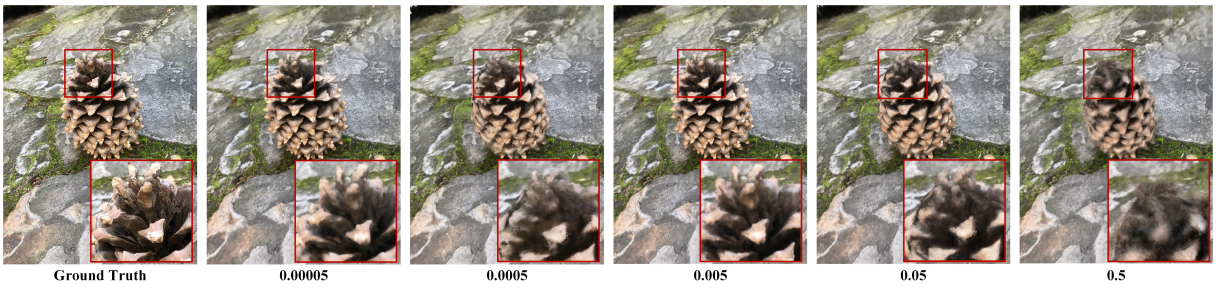

Figure B1 Rendering Results under Different Sampling Ranges. We alter the sampling factor to change the sampling radius and visualize the rendering results at different radii. Compared to the ground truth, the highest fidelity is achieved when the sampling factor is set to 0.005.

Table B1 Quantitative Comparison under Different Sampling Ranges. When the sampling factor $p = 0.005$, PD-NeRF outperform all evaluation metrics and do not consume excessive training time compared to scenarios without pseudo-depth supervision.

Factor p	PSNR↑	SSIM↑	LPIPS↓	times
0.00005	20.224	0.664	0.813	**2.0mins**
0.00050	19.026	0.634	0.835	2.5mins
0.00500	**21.623**	**0.711**	**0.802**	**2.7mins**
0.05000	19.052	0.634	0.843	3.5mins
0.50000	18.446	0.620	0.863	5.0mins

In fact, we do not directly use f to control the sampling rate. Instead, we control the sampling range by applying a sampling radius r to the feature points, where $r = p \times$ image_width and p is the sampling factor. f is merely a scaling factor that changes

with the image width. For a 1600x1600 pixel image, COLMAP typically detects around 533 feature points per image, resulting in approximately 533 pixels with true depth. Given that we set $\omega_{ri} \leqslant 0.01, \omega_{ri} = 0$, and for $\omega_{ri} = 0.01$, with $f = 1$ the Gaussian distribution's sampling radius r is approximately 3.03, meaning each feature point covers around 28.84 pixels. After Gaussian sampling, the total sampled pixels account for 0.6% of the image's pixel count ($28.84 \times 533/(1600 \times 1600) \approx 0.0006$ and 6% mentioned in the paper is incorrect). The radius of 3.03 corresponds to approximately 0.002 times the image width of 1600. In our experiments, we used a slightly larger value of 0.005. As the image resolution changes, the number of feature points will increase proportionally, and the sampling radius $r = 0.005 \times$ image_width, with the scaling factor f adjusting to $\frac{1}{1600} \times$ image_width. Therefore, the actual sampling coverage of 0.6% is determined by the sampling radius r. We will compare the effect of different sampling radii, i.e., varying p, on the rendering results.

We conducted ablation experiments with five different levels of sampling factors. In Figure B1, we report the rendering results after 10,000 iterations under varying sampling ranges. When the sampling factor is set to 0.005, PD-NeRF achieves the highest fidelity. If the sampling factor is too large, the confidence of the pseudo-depth becomes too low, resulting in degraded rendering quality. Conversely, if the sampling range is too small, the depth supervision becomes insufficient, yielding results similar to the original Instant-NGP [3]. In Table B1, we present qualitative results. A larger sampling range also increases the computational load; for instance, when the sampling factor is set to 0.5, the training time for 10,000 iterations more than doubles. With a sampling factor of 0.005, we achieve the best rendering quality while incurring minimal additional time costs.

References

1 Deng K, Liu A, Zhu J Y, et al. Depth-supervised NeRF: Fewer views and faster training for free. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, 2022. 12882-12891

2 Mildenhall B, P.Srinivasan P, Tancik M, et al. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In: Proceedings of European conference on computer vision, Glasgow, 2020. 405-421

3 Müller T, Evans A, Schied C, et al. Instant neural graphics primitives with a multiresolution hash encoding. ACM Transactions on Graphics (ToG), 2022, 41.4: 1-15