Supplementary Material for "UHDNeRF: Ultra-High-Definition Neural Radiance Fields"

Quewei Li[1], Feichao Li[1], Jie Guo[1,†] and Yanwen Guo[1,†]
[1]National Key Lab for Novel Software Technology, Nanjing University, China
{queweili, feichaoli}@smail.nju.edu.cn, {guojie, ywguo}@nju.edu.cn

1. Introduction

In this supplementary material, we detail the pipeline of our UHDNeRF, *i.e.*, regressing sampled locations to the scene properties with a frequency separation strategy. Furthermore, we provide a more detailed evaluation on the LLFF dataset both qualitatively and quantitatively. In addition, we offer a supplementary video on representative scenes (leaves and flowers) at resolution 4032×3024 to better illustrate the superiority of our method on 4K rendering.

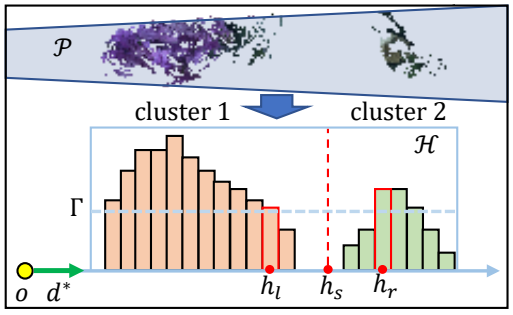


Figure 1. Pipeline of clustering.

2. Details of UHDNeRF

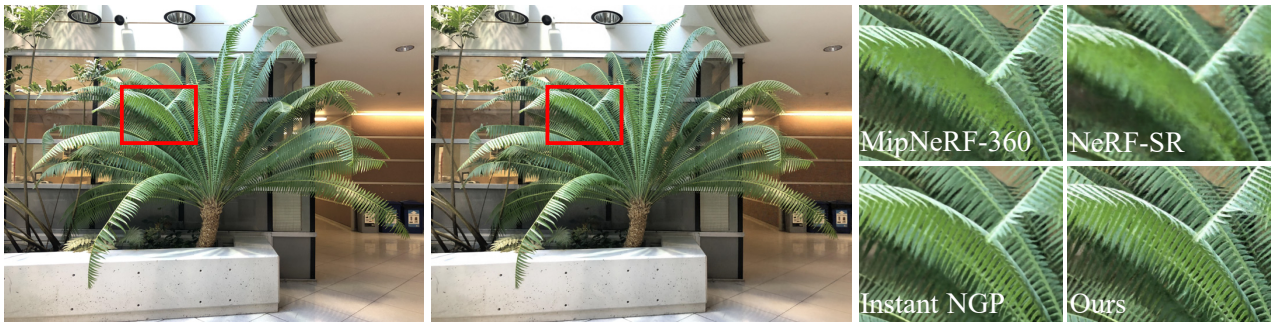
In this section, we detail the pipeline of our UHDNeRF. As described in algorithm 1, the inputs of UHDNeRF are the set of N samples $\mathbf{X}$ sampled from rays within a local patch, the corresponding viewing directions $\mathbf{D}$, the point cloud $\mathbf{P}$ containing $\bar{N}$ 3D points, and the number of neighboring points k per sample. We first query k nearest neighbors (we only store the point indexes for saving memory) for each sample in $\mathbf{X}$ (Line 2) with the fast query method introduced in Point-NeRF [1]. The k points of a sample $\boldsymbol{x}$ are sorted from near to far, and if there are no more neighboring points, we set NULL in the corresponding entries of $\mathcal{P}_{ind}$. Since $\mathcal{P}_{ind}$ is redundant (a point may appear several times), we calculate the unique point set $\mathcal{P}$ (containing M points, where $M \ll N$) and generate the corresponding point feature set $\mathcal{F}$ (Line 3∼4).

Algorithm 1: Pipeline of UHDNeRF

Input: the set of sample locations $\mathbf{X} \in \mathbb{R}^{N\times 3}$, the directions $\mathbf{D} \in \mathbb{R}^{N\times 3}$, the point cloud $\mathbf{P} \in \mathbb{R}^{\bar{N}\times 6}$, and the neighbor number k.
Output: the sets of densities $\mathbf{V} \in \mathbb{R}^{N}$ and colors $\mathbf{C} \in \mathbb{R}^{N\times 3}$.

1 $\mathbf{V} \leftarrow \{\}, \mathbf{C} \leftarrow \{\}$;
2 $\mathcal{P}_{ind} \in \mathbb{R}^{N\times k} \leftarrow$ **query_point_Index**$(\mathbf{X}, \mathbf{P}, k)$;
3 $\mathcal{P} \in \mathbb{R}^{M\times 6} \leftarrow$ **get_unique_points**$(\mathbf{P}, \mathcal{P}_{ind})$;
4 $\mathcal{F} \in \mathbb{R}^{M\times 64} \leftarrow \Phi_{point}(\mathcal{P})$;
5 $\zeta \in \mathbb{R}^{M} \leftarrow$ **cluster**$(\mathcal{P})$;
6 **for** $\boldsymbol{x} \in \boldsymbol{X}$ and $\boldsymbol{d} \in \boldsymbol{D}$ **do**
7 $\quad(\sigma, \mathcal{F}_\sigma) \leftarrow \Phi_\sigma(\boldsymbol{x})$;
8 $\quad\mathbf{V}.\text{add}(\sigma)$;
9 $\quad\boldsymbol{p}_{near} \leftarrow \mathcal{P}[\mathcal{P}_{ind}[\boldsymbol{x}, 0]]$;
10 $\quad$**if** $p_{near} = \text{NULL}$ **then**
11 $\quad\quad\boldsymbol{c} \leftarrow \Phi_L(\mathcal{F}_\sigma, \boldsymbol{d})$;
12 $\quad\quad\mathbf{C}.\text{add}(\boldsymbol{c})$;
13 $\quad$**else**
14 $\quad\quad\mathcal{P}' \in \mathbb{R}^{N'\times 6} \leftarrow \mathcal{P}[\zeta = \zeta[\boldsymbol{p}_{near}]]$;
15 $\quad\quad\mathcal{F}_p \in \mathbb{R}^{N'\times 64} \leftarrow \mathcal{F}[\mathcal{P}']$;
16 $\quad\quad\mathcal{F}_{g,x} \in \mathbb{R}^{64} \leftarrow \Phi_{global}(\text{Maxpool}(\mathcal{F}_p), \boldsymbol{x})$;
17 $\quad\quad i \leftarrow 0, \mathcal{F}_l \leftarrow \{\}$;
18 $\quad\quad$**while** $\mathcal{P}_{ind}[\boldsymbol{x}, i] \neq \text{NULL}$ **do**
19 $\quad\quad\quad\boldsymbol{p}^i \leftarrow \mathcal{P}[\mathcal{P}_{ind}[\boldsymbol{x}, i]]$;
20 $\quad\quad\quad(\boldsymbol{\gamma}^i, \boldsymbol{\beta}^i) \leftarrow \text{CFT}(\boldsymbol{x}, \boldsymbol{x} - \boldsymbol{p}^i)$;
21 $\quad\quad\quad\mathcal{F}_l.\text{add}(\boldsymbol{\gamma}^i \times \mathcal{F}_p[\boldsymbol{p}^i] + \boldsymbol{\beta}^i)$;
22 $\quad\quad$**end**
23 $\quad\quad\mathcal{F}_{l,x} \in \mathbb{R}^{64} \leftarrow \text{Maxpool}(\mathcal{F}_l)$;
24 $\quad\quad\boldsymbol{c} \leftarrow \Phi_H(\mathcal{F}_\sigma, \boldsymbol{d}, \mathcal{F}_{g,x}, \mathcal{F}_{l,x})$;
25 $\quad\quad\mathbf{C}.\text{add}(\boldsymbol{c})$;
26 $\quad$**end**
27 $\quad$**Return V , C**;
28 **end**

[†]Corresponding authors.



(a) GT (b) Ours (c) Closeups

Figure 2. Comparison with state-of-the-art methods on the fern dataset.

We cluster $\mathcal{P}$ into several groups (Line 5). Since $\mathcal{P}$ is constrained within a cone cast from the camera center $\boldsymbol{o}$ along the central viewing direction $\boldsymbol{d}^*$, we adopt an efficient statistic-based method for clustering. As shown in Fig. 1, we project all the points in $\mathcal{P}$ onto a line parallel to $\boldsymbol{d}^*$. We divide the line segment containing points into t intervals and count the number of projected points in each interval, generating a histogram $\mathcal{H}$. Then, we search for splitting locations of $\mathcal{H}$, which are later used for clustering. Specifically, a splitting location h_s is defined as $h_s = (h_l + h_r)/2$, where h_l is the last histogram bin (beyond a threshold Γ) of a series of adjacent bins, and h_r is the first bin beyond Γ after h_l (see the red rectangles/points in Fig. 1). We utilize all the splitting locations to cluster the point set $\mathcal{P}$, generating a set of cluster numbers ζ. The interval number t is set to 256, and the threshold Γ is half of the highest histogram bin.

For any $\boldsymbol{x} \in \mathbf{X}$, we estimate the density σ at that location and query its nearest neighboring point $\boldsymbol{p}_{near}$ (Line7∼9). If $\boldsymbol{p}_{near}$ does not exist, we directly regress the color $\boldsymbol{c}$ at that location with the low-frequency branch Φ_L (Line 10∼12). Otherwise, we obtain $\boldsymbol{x}$'s point cluster $\mathcal{P}'$ (containing N' points where $N' \leq M$) by gathering points in $\mathcal{P}$ that have the same cluster number as $\boldsymbol{p}_{near}$ (Line 14). We generate the global structure feature $\mathcal{F}_{g,x}$ from $\mathcal{P}'$ (Line 15∼16) and the local point-wise feature $\mathcal{F}_{l,x}$ from the i ($i \leq k$) neighboring points of $\boldsymbol{x}$ (Line 17∼23). After that, we regress the color at $\boldsymbol{x}$ with the high-frequency branch Φ_H (Line 24∼25).

3. More comparisons on the LLFF dataset

The LLFF dataset consists of 8 forward-facing scenes with training views between 20 and 62. All the image resolution is 4032×3024. We hold out $1/8$ of the images for each scene to form the test set as the original NeRF does [2]. Table 1 shows the per-scene quantitative results of the comparisons on the LLFF dataset. As seen, our method achieves the best PSNRs, SSIMs, and LPIPSs in all scenes. Furthermore, we present more visual comparisons with the state-of-the-art methods [3, 4, 5]. As shown in Fig. 2∼8, the proposed UHDNeRF outperforms these methods on generating 4K ultra-high-resolution (4032×3024) results with rich details.

References

[1] Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-nerf: Point-based neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5438–5448, 2022.

[2] 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 *European conference on computer vision*, pages 405–421. Springer, 2020.

[3] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5470–5479, 2022.

[4] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, July 2022.

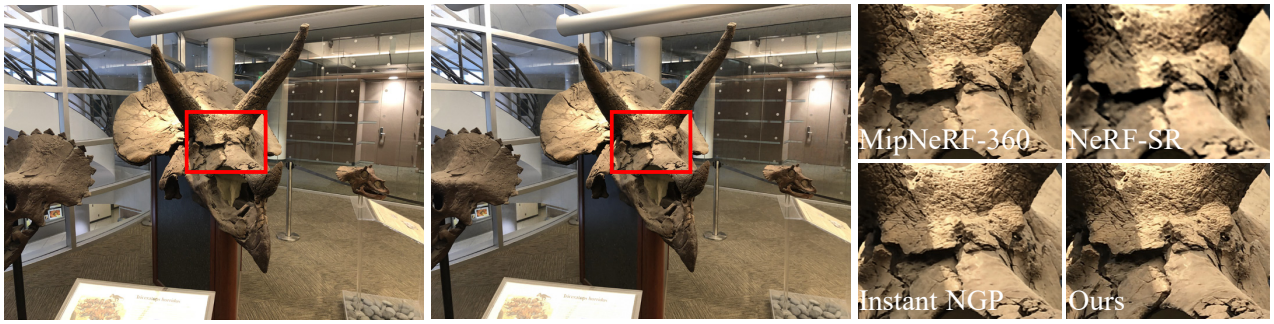
[5] Chen Wang, Xian Wu, Yuan-Chen Guo, Song-Hai Zhang, Yu-Wing Tai, and Shi-Min Hu. Nerf-sr: High quality neural radiance fields using supersampling. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 6445–6454, 2022.

Table 1. Quantitative comparison on the LLFF dataset. We use MipNeRF-360+ and Instant NGP+ to denote the corresponding enhanced versions. The best metrics are highlighted in bold.

Method	Dataset							
	Fern	Flower	Fortress	Horns	Leaves	Orchids	Room	Trex
PSNR ↑								
MipNeRF-360 [3]	23.27	27.07	28.68	25.29	19.23	19.31	29.93	24.96
MipNeRF-360+	23.34	27.13	28.77	25.33	19.35	19.39	30.02	25.05
Instant NGP [4]	25.84	27.62	30.99	27.04	19.81	23.99	33.23	27.92
Instant NGP+	26.32	28.41	31.49	27.64	20.52	24.53	33.70	28.31
NeRF-SR [5]	22.53	25.36	28.33	23.19	18.37	18.45	29.01	23.78
Ours	**27.62**	**30.03**	**32.55**	**28.47**	**22.95**	**26.35**	**34.86**	**29.32**
SSIM ↑								
MipNeRF-360 [3]	0.771	0.797	0.866	0.777	0.636	0.654	0.918	0.815
MipNeRF-360+	0.776	0.805	0.869	0.782	0.645	0.660	0.921	0.821
Instant NGP [4]	0.812	0.806	0.878	0.804	0.691	0.740	0.926	0.840
Instant NGP+	0.819	0.811	0.883	0.809	0.696	0.748	0.929	0.847
NeRF-SR [5]	0.729	0.732	0.818	0.725	0.541	0.618	0.880	0.751
Ours	**0.832**	**0.828**	**0.897**	**0.818**	**0.725**	**0.760**	**0.941**	**0.859**
LPIPS ↓								
MipNeRF-360 [3]	0.414	0.409	0.337	0.435	0.443	0.460	0.341	0.418
MipNeRF-360+	0.388	0.385	0.323	0.417	0.428	0.452	0.335	0.402
Instant NGP [4]	0.366	0.403	0.314	0.396	0.432	0.383	0.316	0.330
Instant NGP+	0.353	0.398	0.301	0.382	0.426	0.378	0.303	0.321
NeRF-SR [5]	0.571	0.594	0.568	0.607	0.616	0.647	0.461	0.517
Ours	**0.308**	**0.340**	**0.283**	**0.359**	**0.364**	**0.359**	**0.287**	**0.308**

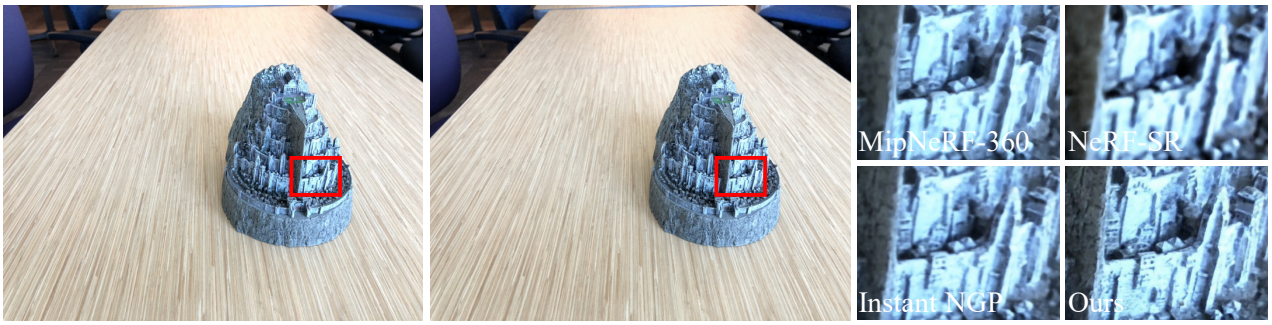
(a) GT (b) Ours (c) Closeups

Figure 3. Comparison with state-of-the-art methods on the flower dataset.



(a) GT (b) Ours (c) Closeups

Figure 4. Comparison with state-of-the-art methods on the horns dataset.

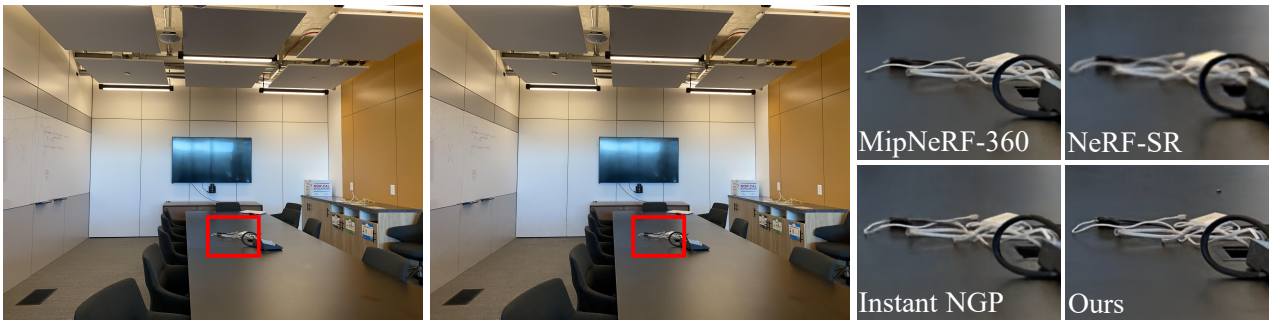


(a) GT (b) Ours (c) Closeups

Figure 5. Comparison with state-of-the-art methods on the fortress dataset.

(a) GT (b) Ours (c) Closeups

Figure 6. Comparison with state-of-the-art methods on the orchids dataset.



(a) GT (b) Ours (c) Closeups

Figure 7. Comparison with state-of-the-art methods on the room dataset.

(a) GT (b) Ours (c) Closeups

Figure 8. Comparison with state-of-the-art methods on the trex dataset.