Appendix – Point-NeRF: Point-based Neural Radiance Fields

Qiangeng Xu[1] † Zexiang Xu[2] Julien Philip [2] Sai Bi[2] Zhixin Shu[2]
Kalyan Sunkavalli[2] Ulrich Neumann[1]
[1]University of Southern California [2]Adobe Research
{qiangenx,uneumann}@usc.edu {zexu,juphilip,sbi,zshu,sunkaval}@adobe.com

A. Ablation Studies on Point Features Initialization

	Extract_{20k}	Rand_{20k}	Extract_{200k}	Rand_{200k}
PSNR↑	**30.71**	25.44	**33.77**	32.01
SSIM↑	**0.967**	0.932	**0.973**	0.972

Table 1. Comparisons between using the extracted image features to initialize the point features (our full model) or using the random initialized features.

We conduct experiments to demonstrate the importance of our feature initialization. We compare our full model and our model initialized without using the extracted image features on the NeRF Synthetic dataset [10]. Without using the features from images, we randomly initialize the point features by using the popular Kaiming Initialization [4]. As shown in Table 1, the neural points with image features not only achieve better performance after convergence at $200K$ iterations but also converge much faster in the beginning. The randomly initialized neural points even cannot perform as well as our full model, still outperforms state-of-the-art methods such as NeRF and NSVF [7].

B. Per-scene Breakdown Results of the DTU Dataset

We show the per scene detailed quantitative results of the comparisons on the DTU [5] dataset in Table 2 and additional qualitative comparisons in our video. Since our method also faithfully reconstructs the scene geometry, our method has the best SSIM scores in most of the cases. Our model also has the best LPIPS for most of the scenes and therefore, is more visually authentic, as shown in the Figure 6 of the main paper and the video. IBRNet combines the colors from the source views to compute the radiance colors during shading. This image-based approach results in better PSNR. However, as shown in our video, our method is more temporal consistent because the lo-

Scan	#1	#8	#21	#103	#114
SSIM↑					
Ours_{1K}	0.935	0.906	0.913	0.944	0.948
Ours_{10K}	0.962	0.949	0.954	0.961	0.960
MVSNeRF_{10K} [2]	0.934	0.900	0.922	0.964	0.945
IBRNET_{10K} [14]	0.955	0.945	0.947	0.968	0.964
NeRF_{200K} [10]	0.902	0.876	0.874	0.944	0.913
$\text{LPIPS}_{Vgg}\downarrow$					
Ours_{1K}	0.151	0.207	0.201	0.208	0.148
Ours_{10K}	0.095	0.130	0.134	0.145	0.096
MVSNeRF_{10K}	0.171	0.261	0.142	0.170	0.153
IBRNET_{10K}	0.129	0.170	0.104	0.156	0.099
NeRF_{200K}	0.265	0.321	0.246	0.256	0.225
PSNR↑					
Ours_{1K}	28.79	28.39	24.78	30.36	29.82
Ours_{10K}	30.85	30.72	26.22	32.08	30.75
MVSNeRF_{10K}	28.05	28.88	24.87	32.23	28.47
IBRNET_{10K}	31.00	32.46	27.88	34.40	31.00
NeRF_{200K}	26.62	28.33	23.24	30.40	26.47

Table 2. Quantity comparison on five sample scenes in the DTU testing set with the view synthesis setting introduced in [2]. The subscripts indicate the number of iterations during optimization.

cal radiance and geometries are consistently stored at each neural point location.

C. Per-scene Breakdown Results of the NeRF Synthetic Dataset

We show the per scene detailed quantitative results of the comparisons on the NeRF Synthetic [10] dataset in Table 3 and additional qualitative comparisons in our video. Point-NeRF achieves the best PSNRs, SSIMs and LPIPSs on most of the scenes and outperforms state-of-the-art methods [1, 7, 10, 14] with a big margin. On the other hand, our method initiated with COLMAP points is on par with NeRF. Even starting from the unideal initial points, we still manage to improve the geometry reconstruction and generate a high-quality radiance field with point pruning and growing. The fact that our model at $20K$ iterations matches the results of NeRF at $500K$ iterations clearly demonstrates our ability of fast convergence.

D. Evaluation on Large-scale 3D Scenes (ScanNet).

While our model is purely trained on a dataset of objects (the DTU dataset), our network generalizes well to large-scale 3D scene datasets. Following [7], we use two 3D scenes, scene 0101_04 and scene 0241_01, from ScanNet [3]. We extract both RGB and depth images from the original videos and from which we sample one out of five frames as training set and use the rest for testing. The RGB images are scaled to 640×480. We finetune each scene for 300K steps with point pruning and growing.

We compare with 3 other state-of-the-art methods with quantitative results in Tab. **??**. In particular, we compare with a scene representation model (SRN) [13], NeRF [10] and a sparse voxel-based neural radiance field, NSVF [7]. The qualitative comparison is shown in Tab. 4 and visual results are shown in Figure 1. Our Point-NeRF outperforms all these previous studies in all metrics by substantial margins. Please find more visual results in our video.

E. The Tanks and Temple Dataset

We also experiment Point-NeRF on the Tanks and Temples dataset [6]. we reconstruct the radiance field of five scenes selected in NSVF [7] and compare our model with three models NV [8], NeRF [10] and NSVF [7]. We show the quantitative comparison in Tab. 5 and visualize quality results in Figure 2. Please find more visual results in our video.

F. Initializing Neural Points from COLMAP Points

Point-NeRF can use the points of any external reconstruction method. For instance, the output of COLMAP [12] is a point cloud $\{(p_i)|i = 1, ..., N\}$. We set γ_i as 0.3 in the beginning. The confidence score of valid points will be pushed to 1 during the optimization process. To acquire point features f_i for a point, We first rule out all the views where the point is occluded by other points, then we find the view of which the camera is the closest to the point. Then from that view, we can unproject the point onto the feature maps extracted by G_f (see Figure 2(a) in the main paper) from the selected view and obtain the f_i.

G. Networks Architectures

Cost volume-based CNN $G_{p,\gamma}$. Our cost volume-based CNN adopts the popular architecture of [17], which is simple and efficient. It includes three layers of depth features extraction CNN, while the latter two layers down-samples the spatial dimension by 4 and output a feature map with 32 channels. Then, these features from each view will be warped according to camera pose and the variance will be computed. The variance features will go through a narrow U-Net [15] and output a 1-channel feature to calculate the depth probability.

Image Feature Extraction 2D CNN G_f. The image feature extraction network takes inputs of RGB image and has three down-sampling layers, each output feature with channels of $8, 16, 32$. We extract the point features by unprojecting a 3D point to each layer and taking the multi-scale features.

	NeRF Synthetic							
	Chair	Drums	Lego	Mic	Materials	Ship	Hotdog	Ficus
	PSNR↑							
NPBG [1]	26.47	21.53	24.84	26.62	21.58	21.83	29.01	24.60
NeRF [10]	33.00	25.01	32.54	32.91	29.62	28.65	36.18	30.13
NSVF [7]	33.19	25.18	32.54	34.27	**32.68**	27.93	37.14	31.23
Point-NeRF$^{col}_{200K}$	35.09	25.01	32.65	35.54	26.97	30.18	35.49	33.24
Point-NeRF$_{20K}$	32.50	25.03	32.40	32.31	28.11	28.13	34.53	32.67
Point-NeRF$_{200K}$	**35.40**	**26.06**	**35.04**	**35.95**	29.61	**30.97**	**37.30**	**36.13**
	SSIM↑							
NPBG	0.939	0.904	0.923	0.959	0.887	0.866	0.964	0.940
NeRF	0.967	0.925	0.961	0.980	0.949	0.856	0.974	0.964
NSVF	0.968	0.931	0.960	0.987	**0.973**	0.854	0.980	0.973
Point-NeRF$^{col}_{200K}$	0.990	0.944	0.983	0.993	0.955	0.941	0.986	0.989
Point-NeRF$_{20K}$	0.981	0.944	0.980	0.986	0.959	0.916	0.983	0.986
Point-NeRF$_{200K}$	**0.991**	**0.954**	**0.988**	**0.994**	0.971	**0.942**	**0.991**	**0.993**
	LPIPS$_{Vgg}$ ↓							
NPBG	0.085	0.112	0.119	0.060	0.134	0.210	0.075	0.078
NeRF	0.046	0.091	0.050	0.028	0.063	0.206	0.121	0.044
Point-NeRF$^{col}_{200K}$	0.026	0.099	0.031	0.019	0.100	0.134	0.061	0.028
Point-NeRF$_{20K}$	0.051	0.103	0.054	0.039	0.102	0.181	0.074	0.043
Point-NeRF$_{200K}$	**0.023**	**0.078**	**0.024**	**0.014**	**0.072**	**0.124**	**0.037**	**0.022**
	LPIPS$_{Alex}$ ↓							
NSVF	0.043	0.069	0.029	0.010	**0.021**	0.162	0.025	0.017
Point-NeRF$^{col}_{200K}$	0.013	0.073	0.016	0.011	0.076	0.087	0.032	0.012
Point-NeRF$_{20K}$	0.027	0.057	0.022	0.024	0.076	0.127	0.044	0.022
Point-NeRF$_{200K}$	**0.010**	**0.055**	**0.011**	**0.007**	0.041	**0.070**	**0.016**	**0.009**

Table 3. Detailed breakdown of quantitative metrics of individual scenes for the NeRF Synthetic [10] for our method and baselines. All scores are averaged over the testing images. The subscripts are the number of iterations of the models and Point-NeRF$^{col}_{200K}$ indicates our method initiates from COLMAP points and optimized for 200 thousand iterations.

	Average over two scenes				**Scene 101**	**Scene 241**
	SRN [13]	NeRF [9]	NSVF [7]	Point-NeRF (Ours)	Point-NeRF (Ours)	
PSNR ↑	18.25	22.99	25.48	**30.32**	30.13	30.51
SSIM ↑	0.592	0.620	0.688	**0.909**	0.912	0.906
RMSE ↓	14.764	0.681	0.079	**0.031**	0.032	0.030
LPIPS$_{Alex}$ ↓	0.586	0.369	0.301	**0.220**	0.203	0.238
LPIPS$_{Vgg}$ ↓	-	-	-	**0.292**	0.286	0.299

Table 4. Quantity comparison on two scenes in the ScanNet dataset [3] selected in NSVF [7]. RMSE is the Root Mean Square Error. Our method Point-NeRF outperforms all state-of-the-art methods in all metrics by substantial margins.

Point-based Radiance Fields MLP. We visualize the details of the point feature aggregation and radiance computation in Figure 3. In all of our experiments, we set $c_1 = 56$, $c2 = 128$. The MLPs F, R, T have 2, 3, 2 layers, respectively. The intermediate feature channels of F and T are 256, and

Figure 1. The qualitative results of our Point-NeRF on the ScanNet dataset [6]. The first row shows five generated test frames of scene 101 and the second row shows five generated test frames of scene 241.

	Tanks & Tamples					
	Ignatius	Truck	Barn	Caterpillar	Family	Mean
			PSNR ↑			
NV [8]	26.54	21.71	20.82	20.71	28.72	23.70
NeRF [10]	25.43	25.36	24.05	23.75	30.29	25.78
NSVF [7]	27.91	26.92	27.16	26.44	33.58	28.40
Point-NeRF (Ours)	**28.43**	**28.22**	**29.15**	**27.00**	**35.27**	**29.61**
			SSIM ↑			
NV [8]	0.992	0.793	0.721	0.819	0.916	0.848
NeRF [10]	0.920	0.860	0.750	0.860	0.932	0.864
NSVF [7]	0.930	0.895	0.823	0.900	0.954	0.900
Point-NeRF (Ours)	**0.961**	**0.950**	**0.937**	**0.934**	**0.986**	**0.954**
			LPIPS_{Alex} ↓			
NV [8]	0.117	0.312	0.479	0.280	0.111	0.260
NeRF [10]	0.111	0.192	0.395	0.196	0.098	0.198
NSVF [7]	0.106	0.148	0.307	0.141	0.063	0.153
Point-NeRF (Ours)	**0.069**	**0.077**	**0.120**	**0.111**	**0.024**	**0.080**
			LPIPS_{Vgg} ↓			
Point-NeRF (Ours)	**0.079**	**0.117**	**0.180**	**0.156**	**0.046**	**0.115**

Table 5. Quantity comparison on five scenes in the Tanks and Temples dataset [6] selected in NSVF [7]. Our method Point-NeRF outperforms all state-of-the-art models in all metrics by substantial margins.

128 channels for R.

H. Neural Point Querying

To efficiently query neural point neighbors for ray marching, inspired by the CAGQ point query introduced in [16], we implement a grid query method. Then we build grid-point indices which register each neural point to evenly spaced 3D grids. Since these grids in the perspective coordinate are cubic, in the world coordinate, they have shapes of spherical voxels.

With the grid-point indices, we can discover grids that have neural points and also their grid neighbors. These grid neighbors are the regions of interest since there should exist neural points within the query radius. If a ray crosses these regions, we can place shading points inside. Finally, we query neural points by directly retrieving the stored neu-

Figure 2. The qualitative results of our Point-NeRF on the Tanks and Temples dataset.

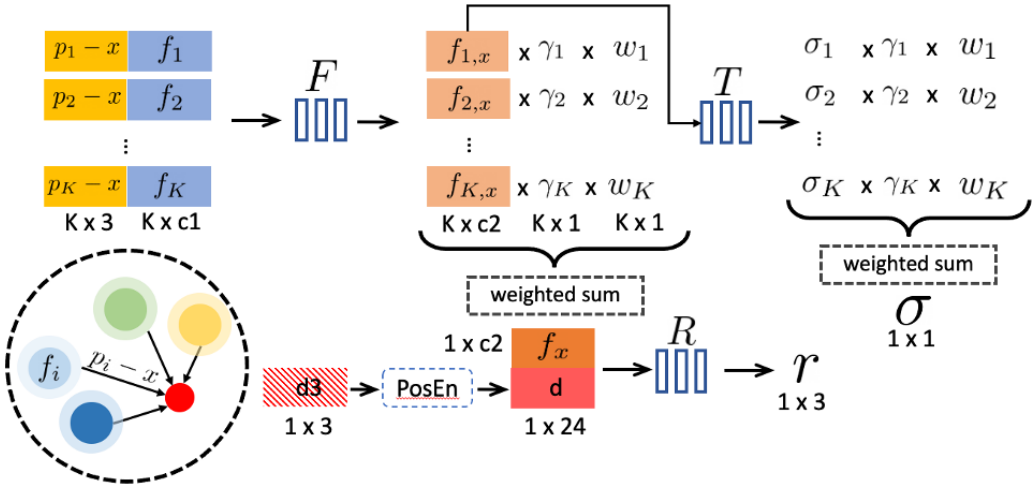

Figure 3. The network pipeline of radiance fields computation at a shading location x from K neural points neighbors. "PosEN" indicates positional encoding [10]. "d3" indicates the 3 channels vector of view directions at x. The final outputs are the radiance color r and density σ. Please also refer to the equations (3-7) in the main paper.

ral points according to the grid-point indices.

In all of our experiments, we query 8 nearest neural point neighbors for each shading location. Along each ray, we only search for neural point neighbors and compute radiance for shading locations in a grid that is occupied itself or nearby occupied grids. Therefore, our shading is much more efficient by skipping the empty space, unlike other radiance fields representations. This is one key advantage that enables fast convergence. Even NSVF [7], high-performance local radiance representation, has to probe the empty space in the beginning and gradually prune the voxels along its training process.

The benefit of this strategy is two-fold: First, we only place shading points in the area that exists neural points, so that we avoid radiance computation in the empty space. Second, the nearby points can be efficiently retrieved according to the indices, which substantially accelerate the point query speed.

I. Limitations

Because we do not focus on the rendering speed and we have not optimized our implementation (point querying and point feature aggregation) for fast rendering. Although, our model is naturally

faster than NeRF (3X) due to that we skip the shading in empty space. We believe future works on combining mechanisms introduced in current papers such as [11, 18] with our point-based radiance representation would further benefit the neural rendering technology.

References

[1] Kara-Ali Aliev, Artem Sevastopolsky, Maria Kolos, Dmitry Ulyanov, and Victor Lempitsky. Neural point-based graphics. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16*, pages 696–712. Springer, 2020. 2, 3

[2] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. *arXiv preprint arXiv:2103.15595*, 2021. 1

[3] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 2, 3

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015. 1

[5] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engil Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *2014 CVPR*, pages 406–413. IEEE, 2014. 1

[6] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4), 2017. 2, 4

[7] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *arXiv preprint arXiv:2007.11571*, 2020. 1, 2, 3, 4, 5

[8] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *arXiv preprint arXiv:1906.07751*, 2019. 2, 4

[9] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7210–7219, 2021. 3

[10] 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. 1, 2, 3, 4, 5

[11] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. *arXiv preprint arXiv:2103.13744*, 2021. 6

[12] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise View Selection for Unstructured Multi-View Stereo. In *European Conference on Computer Vision (ECCV)*, 2016. 2

[13] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. *arXiv preprint arXiv:1906.01618*, 2019. 2, 3

[14] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul Srinivasan, Howard Zhou, Jonathan T. Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *CVPR*, 2021. 1, 2

[15] W Weng and X Zhu. Convolutional networks for biomedical image segmentation. *IEEE Access*, 2015. 2

[16] Qiangeng Xu, Xudong Sun, Cho-Ying Wu, Panqu Wang, and Ulrich Neumann. Grid-gcn for fast and scalable point cloud learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5661–5670, 2020. 4

[17] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. MVSnet: Depth inference for unstructured multi-view stereo. In *Proc. ECCV*, pages 767–783, 2018. 2

[18] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. *arXiv preprint arXiv:2103.14024*, 2021. 6