Supplementary Material for PARF: Primitive-Aware Radiance Fusion for Indoor Scene Novel View Synthesis

Haiyang Ying[1], Baowei Jiang[1], Jinzhi Zhang[1], Di Xu[2], Tao Yu[1†], Qionghai Dai[1], Lu Fang[1†]

[1]Tsinghua University, [2]Huawei Cloud

We provide more implementation details for the proposed primitive-aware radiance fusion method (PARF) for indoor scene novel view synthesis and report more experimental results to verify the effectiveness of PARF, including faster convergence, better extrapolation ability, and robustness of sparsity observation.

1. Details of method

1.1. Details of representation

We give more details about the proposed primitive-aware hybrid representation as follows. To represent the scene, we first normalize the scene into a unit volume (with the size of 1.0^3) with an expanding factor $c = 1.2$ to moderately enlarge the volume, which acts like AABB factor defined in InstantNGP [2]. Then the semantic volume $F_{\mathbf{\Theta}}$ can fully cover the scene. During the training stage, the maximum of marching steps on each ray is 1024, and the sampling distance between adjacent samples within D-voxels is fixed as $\sqrt{3}/1024$. For P-voxels, the distance is fixed as a larger value as $\delta_p = 1.0$ for all the experiments. Since a larger δ_i provides a larger opacity $\alpha_i = 1 - exp(-\sigma_i \delta_i)$, which helps decrease the geometry ambiguity at regions of primitives, this setting accelerates the convergence of the primitive regions. Faster convergence in primitive-based regions also means that rendering errors in non-primitive regions account for a greater proportion of the total loss, allowing non-primitive regions to be optimized more quickly. This setting helps achieve rendering primitive and non-primitive regions in a unified manner.

Additional illustration of sampling *P-voxels*. When applying ray marching within a P-voxel, we first sample the ray-plane intersection point. Then the ray marches a step with size ψ (along the plane normal) to sample the next point. The design of ψ prevents samples behind the plane from jumping back to the intersection and being caught in an endless loop. Specifically, when the current marching point $\mathbf{x}$ locates in a P-voxel and $\mathbf{x}$ is behind the corresponding plane $\mathbf{p}$, if the distance between $\mathbf{x}$ and the plane is larger than ψ along the plane normal, we apply dense sampling (as D-voxel) to the current P-voxel.

1.2. Details of framework

After merging the primitive detection results at frame t into the global primitive list $\mathbf{P}_G^{t-1}$, we check if any planes have been removed from $\mathbf{P}_G^{t-1}$. If so, all P-voxels labeled with v_i in the semantic volume $\mathbf{V_s}$ will be set to D-voxels. After that, the new semantic frame t will be fused into the semantic volume $\mathbf{V_s}$.

When executing primitive fusion, pixels labeled as primitive in I_S^t are considered to have higher priority than non-primitive pixels. Specifically, E-voxels assigned by $I_S^t(u_i) > 0$ will be marked with a counter and cannot be changed by pixels labeled as non-primitive (i.e., $I_D^{t+q}(u_i) = 0, q >= 1$) in the following frames. This helps avoid the situation that detection fails in one frame and some voxels in front of a plane are assigned as D-voxels, which may cause floater around the plane.

During primitive fusion, if one voxel has been assigned as a P-voxel with more than one primitive, we take it as a D-voxel and apply volume rendering within it since it may be a voxel lying on the intersection of two varied primitives.

1.3. Hyper-parameters

For the positional encoder, we follow the same parameter definition of multi-resolution hashing in InstantNGP [2]. There are 16 levels of hash grid with resolution varying from 16^3 to 1024^3. The length of each feature embedding is 2, and the hash map size of each level is 2^{19}. For the direction encoder, the degree of spherical harmonics is 4. The network consists of two MLPs: one MLP $F_{\mathbf{\Theta_1}}$ with 1 layer of 64 neurons and another MLP $F_{\mathbf{\Theta_2}}$ with 2 layers of 64 neurons.

$$(\sigma_i, \mathbf{s}, \mathbf{f}) = F_{\mathbf{\Theta_1}}(\gamma(\mathbf{x}_i)), \tag{1}$$

$$\mathbf{c}_i = F_{\mathbf{\Theta_2}}(\gamma(\mathbf{x}_i), SH(\mathbf{d}_i), \mathbf{f}), \tag{2}$$

where $\mathbf{f}$ is the output feature vector from $F_{\mathbf{\Theta_1}}$. For InstantNGP and NeRF-SLAM [4], we use the same settings of position encoding, direction encoding as well as MLP except

†The corresponding authors are Lu Fang (fanglu@tsinghua.edu.cn, http://www.luvision.net/) and Tao Yu (ytrock@tsinghua.edu.cn).

for the extra semantic head. The size of the semantic volume F_{Θ} is 256^3.

2. Additional experiments and explanation

2.1. Details of performance

Here we show the performance of PARF as well as the comparison with the most relevant algorithm NeRF-SLAM [4]. We state that PARF is a radiance fusion method that can be integrated into SLAM systems for real-time scene reconstruction. PARF consists of primitive detection, merging, fusion, and global optimization. Primitive detection [3] and merging of each depth frame can run at 50 fps and 30fps, respectively, at the resolution of 1200×680 on CPU. Fusing the new semantic frame into the semantic volume can be run at around 30 fps.

We compare the performance of PARF and NeRF-SLAM in Tab. 1, which is also claimed as a real-time SLAM system. All the experiments are run on a single NVIDIA RTX 3090 GPU. For rendering, we apply an accumulated occupancy threshold T for each ray, which means the network inference will stop when the accumulated occupancy has been larger than $(1 - T)$. Therefore, the number of sampled points in ray marching may differ from the number used for volume rendering. We apply $T = 0.0001$ for training and evaluation while using $T = 0.01$ for real-time rendering. We find that PARF achieves faster rendering speed and needs less point sampling for rendering thanks to the sparse modeling based on the proposed primitive-aware hybrid representation.

Index	NeRF-SLAM	PARF
RM (pts/ray)	128	**17.6**
VR (pts/ray)	15.2	**3.5**
Speed 1 (iter/s)	49.5	**116.1**
Speed 2 (fps)	31.3	**62.5**

Table 1: Speed Analysis on Replica dataset. RM: average number of ray marching points on each ray, VR: average number of volume rendering points on each ray, Speed 1: average iterations per second, Speed 2: average render speed at resolution 1200×680.

2.2. Explanation of teaser

Most SLAM systems [1] consist of two parts: a **front-end** for tracking and a **back-end** for reconstruction. In Fig. 1(a) of the manuscript, we assume that 10 new keyframes are sent from the front-end to the back-end per second. In other words, after each second, 10 more frames can be used to detect primitives and optimize the global representation. This guarantees that the frames can be used for optimization are the same for PARF and NeRF-SLAM at the same moment, which provides a fair comparison between PARF and NeRF-SLAM without considering the speed of the front-end. At each moment, all frames that have been fed to the back-end are used to optimize the representations of PARF and NeRF-SLAM. The incremental reconstruction in Fig. 1(a) and the curve in Fig. 7 in the manuscript depict that PARF achieves much faster convergence for free-view synthesis and extrapolation.

NeRF-SLAM Note that the NeRF-SLAM we implement in this work is the mapping stage of the original work [4], which is an InstantNGP accompanied by a depth render loss $\mathcal{L}_d = \sum_{\mathbf{r}} \|d(\mathbf{r}) - d_{\text{gt}}(\mathbf{r})\|_2^2$. Besides, the depth used for supervision is the ground truth depth instead of the predicted depth from the tracking stage.

2.3. Geometry quality evaluation

With the help of primitive-level representation, PARF enjoys much more accurate geometry reconstruction results. We compare L1 depth render error (cm) under both interpolation and extrapolation views. The metrics show that PARF consistently outperforms NeRF-SLAM, which further proves the advantage of the hybrid representation of PARF.

Methods	**Mean**	**Interpolation**	**Extrapolation**
InstantNGP	33.71	38.41	29.01
NeRF-SLAM	2.430	2.387	2.473
PARF	**1.401**	**1.554**	**1.249**

Table 2: Geometry quality analysis. We calculate the L1 error of rendered depth maps with unit cm.

2.4. Ablation study: sparsity

Primitives provide strong prior for scene perception, enabling robust representation and optimization when the observation is relatively sparse. We evaluate PARF and NeRF-SLAM with different sparsity levels of input frames on `office0` of the Replica dataset. The sparsity n of the input frames means we take one of every n images in the original sequence (2000 frames) as input. The full results are shown in Tab. 3 and Fig. 1. In Fig. 1, as the degree of sparsity increases, NeRF-SLAM shows a significant drop in performance, while PARF is more robust to sparser input, especially in the regions near the planes. From Tab. 3, we can find that PARF performs consistently better than NeRF-SLAM. Surprisingly, the LPIPS of PARF at sparsity $n = 140$ is still better than that of NeRF-SLAM at sparsity $n = 20$, which further illustrates the effectiveness of the proposed primitive-aware fusion method for sparse-view reconstruction.

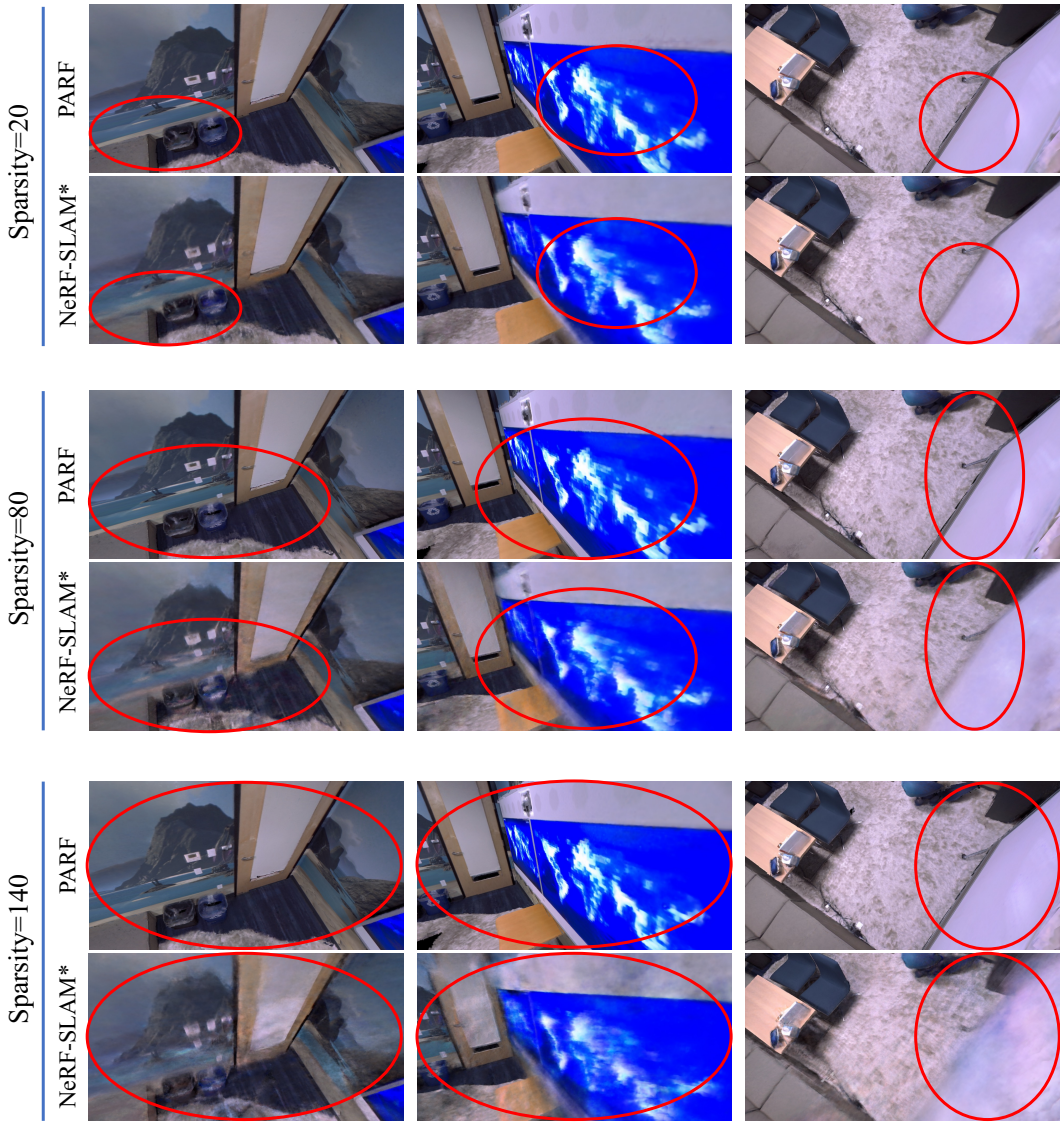

Figure 1: Qualitative comparison of PARF and NeRF-SLAM for different sparsity settings.

Methods	Evaluation	Sparsity						
		20	**40**	**60**	**80**	**100**	**120**	**140**
NeRF-SLAM	PSNR ↑	32.09	31.82	31.27	29.96	28.09	26.79	25.68
	SSIM ↑	0.919	0.909	0.908	0.895	0.879	0.853	0.831
	LPIPS ↓	0.274	0.280	0.287	0.298	0.316	0.352	0.366
PARF	PSNR ↑	**33.18**	**32.80**	**32.67**	**32.28**	**31.53**	**31.24**	**29.76**
	SSIM ↑	**0.934**	**0.920**	**0.923**	**0.916**	**0.911**	**0.913**	**0.905**
	LPIPS ↓	**0.192**	**0.199**	**0.201**	**0.201**	**0.204**	**0.207**	**0.214**

Table 3: Ablation study on sparsity of input frames.

2.5. Ablation study: sampling strategy

This ablation study (in Tab. 4 of the manuscript) aims to evaluate the robustness of our primitive-guided sampling strategy introduced in primitive-aware hybrid representation. Since depth images are available, using depth value for sampling guidance will be more straightforward. How-

Methods	Evaluation	scene0012	scene0027	scene0457	mean
InstantNGP	PSNR ↑	22.60	17.88	22.63	21.04
	SSIM ↑	0.645	0.684	0.726	0.685
	LPIPS ↓	0.573	0.533	0.485	0.530
NeRF-SLAM	PSNR ↑	25.32	20.79	23.74	23.28
	SSIM ↑	0.677	0.734	0.737	0.716
	LPIPS ↓	0.542	0.462	0.464	0.489
PARF	PSNR ↑	**26.03**	**21.54**	**24.22**	**23.93**
	SSIM ↑	**0.689**	**0.739**	**0.746**	**0.725**
	LPIPS ↓	**0.526**	**0.448**	**0.448**	**0.474**

Table 4: Evaluation results on ScanNet dataset.

Methods	Evaluation	apt0	apt2	copyroom	office2	mean
InstantNGP	PSNR ↑	23.20	20.22	20.51	21.75	21.42
	SSIM ↑	0.733	0.648	0.797	0.717	0.724
	LPIPS ↓	0.457	0.480	0.442	0.462	0.460
NeRF-SLAM	PSNR ↑	28.97	22.27	25.34	23.33	24.98
	SSIM ↑	0.772	0.672	**0.827**	0.727	0.749
	LPIPS ↓	0.368	0.439	0.344	0.424	0.394
PARF	PSNR ↑	**29.67**	**23.55**	**26.00**	**24.07**	**25.82**
	SSIM ↑	**0.789**	**0.691**	0.821	**0.741**	**0.760**
	LPIPS ↓	**0.336**	**0.389**	**0.321**	**0.405**	**0.363**

Table 5: Evaluation results on BundleFusion dataset.

ever, depth from sensors inevitably contains noise, which is harmful when using depth as guidance directly. In order to evaluate this, we apply a simple version of depth-guided sampling strategy to NeRF-SLAM. Specifically, if a ray holds a valid depth value on the depth image, we only sample the points located on and behind the depth value during training. Besides, we maintain a simpler version of our proposed semantic volume, which has only D-voxels and E-voxels. In this volume, geometry fusion is implemented by the equation $V^t_{v=0} = \{v_i | I^t_D(u_i) - B_1 < D^t(x_i) < I^t_D(u_i) + B_1\}$. During test time, the sampled points will skip E-voxels and only locates within D-voxels. Since this is a naive way of migrating depth guidance into the volumetric rendering optimization, the density distribution around surfaces will be unstable. Therefore, the sampling and reconstruction performance will drop significantly (as depicted in the Tab. 5 of the manuscript). On the other hand, PARF predicts primitives from sequential observations of the scene, which helps filter noise and construct a more stable scene representation. Note that the sigma of Gaussian noise is 100 mm, which approximates the error of depth sensors.

2.6. Ablation study: Semantic render loss

We add a semantic head and a semantic render loss $\mathcal{L}_s$ to optimize the view-independent semantic information of the scene. The continuous modeling of the semantic field enables the representation of an unlimited number of planes, which has the potential of replacing explicit semantic volume. Besides, since the primitive detection may be noisy on each isolated view, the semantic field can be further rendered to validate and filter noisy planes in the plane list and improve the robustness of the primitive representation and, therefore, improve the rendering performance. Quantitative results show that before and after adding the semantic head to the framework, no apparent performance drop appears (from PSNR: 35.24, SSIM:0.944, LPIPS:0.225 to PSNR: 35.09, SSIM: 0.943, LPIPS: 0.228). This means that the proposed semantic field may enable further study on geometric-level semantic guided scene reconstruction and novel view synthesis at the cost of little burden.

2.7. Detailed quantitative results

We provide detailed quantitative results for per scene in ScanNet (Tab. 4), BundleFusion (Tab. 5), and Replica (Tab. 6) datasets. Note that PARF consistently outperforms other baselines on all three datasets.

3. Limitations

Though PARF shows advantages in fast convergence, high extrapolation quality, convenient scene edition, and obtains SOTA performance on indoor scene reconstruction and novel view synthesis, there are still some limitations.

Methods	Evaluation	office0	office1	office2	office3	office4	room0	room1	room2	mean
DVGO	PSNR ↑	23.66	25.08	18.33	21.38	22.17	18.74	23.13	23.30	21.97
	SSIM ↑	0.802	0.802	0.783	0.818	0.850	0.632	0.781	0.779	0.781
	LPIPS ↓	0.399	0.452	0.503	0.462	0.474	0.594	0.494	0.514	0.487
Plenoxels	PSNR ↑	23.79	25.14	26.79	29.30	29.27	25.60	29.51	30.92	27.54
	SSIM ↑	0.855	0.778	0.905	0.922	0.926	0.807	0.862	0.885	0.867
	LPIPS ↓	0.353	0.418	0.351	0.335	0.374	0.391	0.377	0.363	0.370
NeRF	PSNR ↑	29.90	36.22	28.78	28.41	33.19	27.75	31.62	31.28	30.89
	SSIM ↑	0.867	0.931	0.898	0.887	0.933	0.846	0.894	0.875	0.891
	LPIPS ↓	0.364	0.353	0.389	0.396	0.369	0.341	0.325	0.379	0.365
InstantNGP	PSNR ↑	30.89	34.59	29.77	32.66	34.15	26.42	30.53	32.52	31.44
	SSIM ↑	0.917	0.890	0.914	0.929	0.943	0.795	0.854	0.891	0.892
	LPIPS ↓	0.285	0.383	0.356	0.313	0.327	0.412	0.400	0.354	0.354
TSDF-Fusion	PSNR ↑	27.20	33.21	25.92	25.22	29.68	24.46	27.11	28.35	27.64
	SSIM ↑	0.865	0.926	0.888	0.876	0.920	0.736	0.819	0.832	0.858
	LPIPS ↓	0.349	0.345	0.376	0.381	0.349	0.404	0.414	0.414	0.379
DS-NeRF	PSNR ↑	29.85	36.46	29.42	27.76	34.53	27.95	32.30	31.01	31.16
	SSIM ↑	0.867	0.936	0.895	0.873	0.933	0.835	0.906	0.850	0.887
	LPIPS ↓	0.359	0.347	0.387	0.421	0.374	0.359	0.312	0.401	0.370
Neurmips	PSNR ↑	31.58	37.60	33.17	28.23	36.76	29.82	**34.41**	34.78	33.29
	SSIM ↑	0.908	0.932	0.938	0.887	0.938	0.911	0.937	0.935	0.923
	LPIPS ↓	0.296	0.301	0.287	0.344	0.290	0.266	0.254	0.283	0.290
NeRF-SLAM	PSNR ↑	32.09	**40.27**	33.09	34.55	37.25	30.21	33.52	35.03	34.50
	SSIM ↑	0.919	**0.965**	0.937	0.941	0.956	0.874	0.918	0.931	0.930
	LPIPS ↓	0.274	0.229	0.296	0.272	0.275	0.319	0.305	0.292	0.283
PARF	PSNR ↑	**33.18**	**39.47**	**34.04**	**34.78**	**37.81**	**31.22**	34.31	**35.92**	**35.09**
	SSIM ↑	**0.934**	0.964	**0.945**	**0.943**	**0.960**	**0.914**	**0.940**	**0.948**	**0.943**
	LPIPS ↓	**0.192**	**0.214**	**0.247**	**0.249**	**0.234**	**0.238**	**0.218**	**0.235**	**0.228**

Table 6: Evaluation results on Replica dataset. We report metrics including PSNR (higher is better), SSIM (higher is better), and LPIPS (lower is better). Note that PARF achieves the best numbers in all three metrics.

First, the performance of the proposed primitive-aware representation may be restricted to the resolution of the semantic volume. The resolution of $\mathbf{V_s}$ we use now is 256^3, and the data type is 8-bit unsigned integer, which consumes approximately 15.6Mb GPU memory. Higher resolution will help reduce aliasing at primitive boundaries but consume more memory, which is a problem of trade-off. Besides, replacing the discrete 3D semantic volume with the continuous semantic field may help alleviate this problem.

Second, though PARF can largely reduce the ambiguity of geometry and shows a strong ability of extrapolation with the primitive-aware representation, there may be some unpleasant texture artifacts in some extrapolation views due to the incomplete scene observation. One way of solving this problem is adding global light modeling [5], which may help decompose diffuse color of objects and global environment light from multi-view observation. By introducing global light modeling, our primitive-aware representation can further enable more realistic indoor scene roaming.

References

[1] Angela Dai, Matthias Nießner, Michael Zollöfer, Shahram Izadi, and Christian Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics 2017 (TOG)*, 2017. 2

[2] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022. 1

[3] Pedro F Proença and Yang Gao. Fast cylinder and plane extraction from depth cameras for visual odometry. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6813–6820. IEEE, 2018. 2

[4] Antoni Rosinol, John J Leonard, and Luca Carlone. Nerf-slam: Real-time dense monocular slam with neural radiance fields. *arXiv preprint arXiv:2210.13641*, 2022. 1, 2

[5] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T Barron, and Pratul P Srinivasan. Ref-nerf: Structured view-dependent appearance for neural radiance fields. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5481–5490. IEEE, 2022. 5