Particle Rendering: Implicitly Aggregating Incident and Outgoing Light Fields for Novel View Synthesis

Supplementary Material

6. Network Architecture

This section gives the detailed architecture of the implement functions used in the Particle Rendering (PR) approach, including $\mathcal{F}_i$, $\mathcal{O}$, $\mathcal{F}_o$, $\mathcal{I}$, and $\mathcal{F}_p$.

Implicit Functions: $\mathcal{F}_o$ and $\mathcal{F}_i$. Fig. 6 illustrates a function that maps a 3D location $\mathbf{x}$ to an outgoing light embedding $\mathbf{e}_i$ or an incident light embedding $\mathbf{e}_o$. Initially, the input location $\mathbf{x}$ is transformed into an embedding through HashEncoding [20]. Subsequently, another two-layer MLP with a width of 64 is used to encode $\sigma, \mathbf{e}_o$ for the outgoing field and $\mathbf{e}_i$ for the incident field. The HashEncoding and tiny MLP make it efficient to predict the light fields.

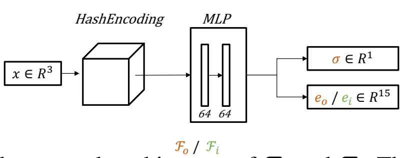

Figure 6. The network architecture of $\mathcal{F}_o$ and $\mathcal{F}_i$. They share the same model network but independent weights.

Implicit Functions: $\mathcal{O}$ and $\mathcal{I}$. The implicit function that maps from outgoing embedding $\mathbf{e}_o$ or incident embedding $\mathbf{e}_i$ to outgoing light $\mathbf{o}$ or incident light $\mathbf{i}$ as demonstrated in Fig. 7. The outgoing embedding $\mathbf{e}_o$ is used as input to three-layer MLPs with a width of 64 to generate the outgoing light o. Similarly, the incident embedding $\mathbf{e}_i$ is fed into another three-layer MLPs with a width of 64 to calculate the incident light i.

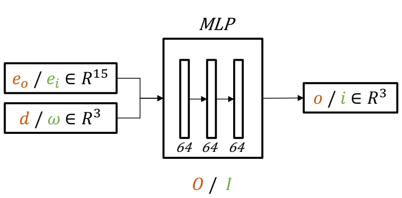

Figure 7. The network architecture of $\mathcal{O}$ and $\mathcal{I}$. They share the same model network but independent weights.

Implicit Function: $\mathcal{F}_p$. Fig. 8 illustrates the mapping from the outgoing light embedding $\mathbf{e}_o$ and incident light embedding $\mathbf{e}_i$ to the final rendered pixel color $\mathbf{C}_p(\mathbf{r})$. These embeddings are combined to form the representation of the camera ray as there are $N = 32$ sampled particles with their respective outgoing and incident embeddings, which is then used to calculate the final color $\mathbf{C}_p(\mathbf{r})$.

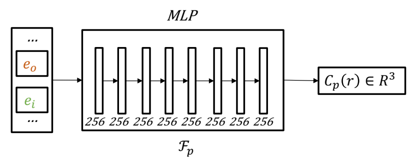

Figure 8. The network architecture of $\mathcal{F}_p$.

Implementation of Implicit Rendering. Various models can be used to execute implicit rendering $\mathcal{F}_p$, such as ConvNet [10], Transformer [25], and MLP [20]. To evaluate their effectiveness and performance, we conducted an experiment, and the results are shown in Tab. 3. We eventually selected MLP [20] as the implementation for implicit rendering because it does not require a lot of resources while achieving satisfactory results.

$\mathcal{F}_p$	PSNR (↑)	Memory (↓)	Training (↓)	Rendering (↑)
Nerfstudio [24]	26.91	4.2GB	0.75 hours	2.2 FPS
Ours w/ ConvNet [10]	29.85	16.2GB	2.0 hours	0.2 FPS
Ours w/ Transformer [25]	**30.24**	20.4BGB	3.0 hours	0.1 FPS
Ours w/ MLP [20]	30.19	**7.9GB**	**0.9 hours**	**2.1 FPS**

Table 3. Comparison between different implementations of Implicit Function $\mathcal{F}_p$.

7. More Experiments

In addition to the *360 Dataset* and *Reflective 360 Dataset*, we also tested our proposed PR on the **Blender Dataset**[19], a synthesized dataset created with the Blender software. Since there is no background information to predict the incident light, our performance is still competitive with the state-of-the-arts, as shown in Tab. 4. This shows the potential of implicit rendering even only considering the outgoing lights.

Method	PSNR ↑	SSIM ↑	LPIPS ↓
NeRF [19]	31.01	0.947	0.081
Instent-NGP [19]	30.00	0.939	0.079
Nerfacto [24]	31.37	0.953	0.072
Mip-NeRF [1]	33.03	**0.964**	**0.037**
Zip-NeRF [3]	**33.46**	**0.962**	0.039
PR w/ outgoing	32.98	0.953	0.049
PR full	**33.11**	0.959	**0.038**

Table 4. Comparison on the collected *Blender Dataset* [19]. PR full does not show visible improvement from PR wo. incident as there is not a significant amount of valid incident light in this scene.

The Structure of Implicit Rendering. For the input of our implicit rendering, there are multiple combinations within the sampled points' attributes, including incident & outgoing embedding $\mathbf{e}$, location $\mathbf{x}$, density σ, direction $\mathbf{d}$, and outgoing light $\mathbf{o}$. In the related work, R2L[27] is similar to ours in implicit rendering. However, they only consider location concatenation for acceleration. We also change the network of $\mathcal{F}_m$ from the Transformer to a Larger MLP with $4\times$ parameters to compare the performance. Tab. 5 shows the ablation about the structure of implicit rendering, and we can observe that our our incident $\&$ outgoing embeddings is enough as input to perform the best performance. Also, we can obverse that larger network cannot improve the performance as implicit renderer.

Table 5. Ablation study of the variants of our implicit rendering.

Method	Ours	$\mathbf{x}_{\text{(R2L[27])}}$	$\mathbf{e}+\mathbf{x}$	$\mathbf{e}+\sigma$	$\mathbf{e}+\mathbf{d}$	$\mathbf{e}+\mathbf{o}$	$4\times$ MLP
PSNR ↑	**30.19**	27.12	30.18	29.85	30.15	29.93	30.03
SSIM ↑	**0.890**	0.855	0.886	0.879	0.886	0.876	0.884
LPIPS ↓	**0.202**	0.262	0.213	0.225	0.207	0.220	0.211

8. More Visualization

We provide additional visual comparisons to demonstrate the superiority of the PR approach over existing methods, as illustrated in Fig. 9 and Fig. 10.

9. Limitation and Future Work.

Since the combination of incident light field and outgoing light fields is not a physical representation, it is thus difficult to edit the scene, *e.g.*, we want to change the intensity and location of light sources and material parameters of the object and render again. It is not yet clear how our incident field directly affects the outgoing field. In our future research, we will thus investigate the implicit representation for light sources and materials apart from particles to analyze the process and develop an implicit approach to render and edit scenes.

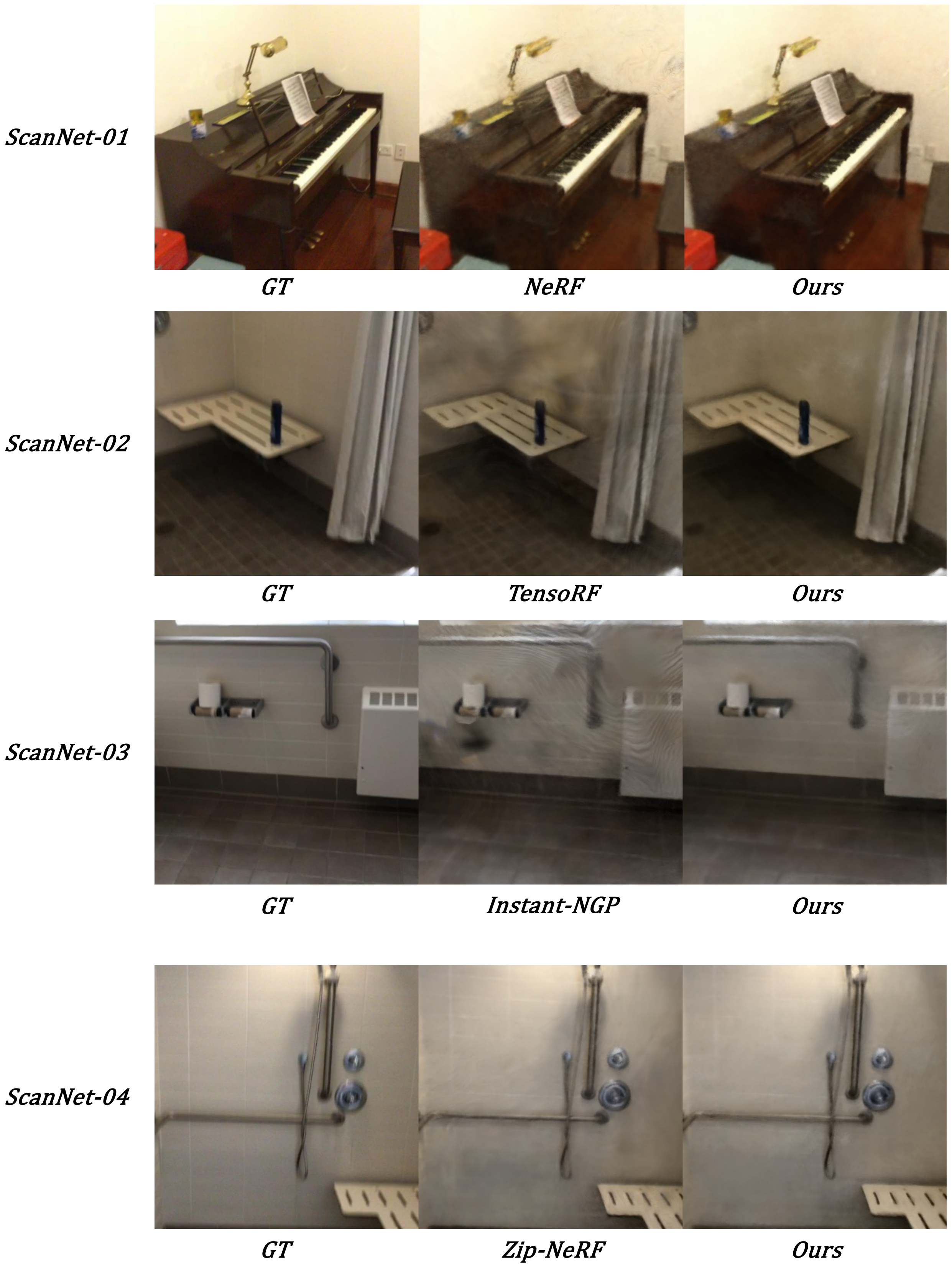

Figure 9. Qualitative comparisons on the ScanNet dataset [7].

Figure 10. Qualitative comparisons on various datasets, including Market from the Mirror-NeRF [33] Dataset, Chamber from our Reflective 360 dataset, and Museum and Ballroom from Tank and Temple dataset [14].