HybridNeRF: Efficient Neural Rendering via Adaptive Volumetric Surfaces

Supplementary Material

A. Color Distillation

We distill the MLP used to represent distance from our 256-wide MLP to a 16-wide network during the finetuning stage (Sec. 3.2). It is possible to further accelerate rendering by similarly distilling the color MLP. We found this to provide a significant boost in rendering speed (from 46 to 60 FPS) at the cost of a minor but statistically significant decrease in rendering quality (see Tab. 5). We observed qualitatively similar results when decreasing width from 64 to 32 channels with more notable changes in color when decreasing the width to 16 channels (see Fig. 9). As our initial results suggest that MLP evaluation remains a significant rendering bottleneck, replacing our scene-wide color MLP with a collection of smaller, location-specific MLPs, as suggested by KiloNeRF [28] and SMERF [7], is potential future work that could boost rendering speed at a smaller cost in quality.

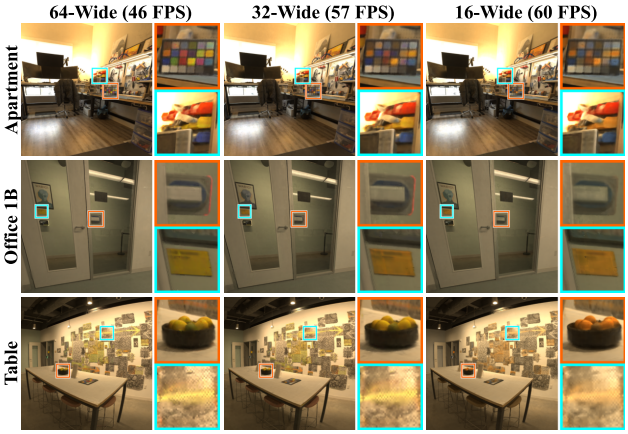

Figure 9. **Color Distillation.** Distilling the color MLP to a smaller width during the finetuning stage (Sec. 3.2) accelerates rendering at the cost of a minor decrease in quality. We observe largely similar results when decreasing the width to 32 channels, and more noticeable changes in color when further decreasing to 16.

Table 5. **Color distillation.** We evaluate the effect of color MLP distillation on the Eyeful Tower dataset [38], and find a significant increase in rendering speed at the cost of quality.

Color Width	↑PSNR	↑SSIM	↓LPIPS	↑FPS
16-wide (distilled)	30.88	0.888	0.236	**60.13**
32-wide (distilled)	<u>31.17</u>	<u>0.900</u>	<u>0.220</u>	<u>57.05</u>
64-wide (original)	**31.57**	**0.913**	**0.198**	45.78

B. Anti-Aliasing

We model rays as cones [1] and use a similar anti-aliasing strategy to VR-NeRF [38] by dampening high-resolution grid features based on pixel footprint. For a given sample $\boldsymbol{x}$, we derive a pixel radius $p(\boldsymbol{x})$ in the contracted space, and calculate the optimal feature level $L(\boldsymbol{x})$ based on the Nyquist–Shannon sampling theorem:

$$L(\boldsymbol{x}) \coloneqq -\log_2(2s \cdot p(\boldsymbol{x})), \tag{7}$$

where s is our base grid resolution (128). We then multiply grid features at resolution level L with per-level weights w_L:

$$w_L = \begin{cases} 1 & \text{if } L < \lfloor L(\boldsymbol{x}) \rfloor \\ L(\boldsymbol{x}) - \lfloor L(\boldsymbol{x}) \rfloor & \text{if } \lfloor L(\boldsymbol{x}) \rfloor < L \leq \lceil L(\boldsymbol{x}) \rceil \\ 0 & \text{if } \lceil L(\boldsymbol{x}) \rceil < L. \end{cases} \tag{8}$$

C. Model architecture

We render color and distance as follows:

$$\boldsymbol{c}(\boldsymbol{x}, \boldsymbol{d}) = \text{MLP}_{\text{col}}(\Gamma_{\text{col}}(\boldsymbol{x}), \text{SH}(\boldsymbol{d})) \tag{9}$$

$$f(\boldsymbol{x}) = \text{MLP}_{\text{dist}}(\Gamma_{\text{dist}}(\boldsymbol{x})), \tag{10}$$

where Γ_{col} and Γ_{dist} are separate spatial feature encodings:

$$\Gamma_\bullet(\boldsymbol{x}) = \bigoplus_{g \in \{G_\bullet, T^1_\bullet, T^2_\bullet, T^3_\bullet\}} \sum_{l=0}^{L_g - 1} w_l \cdot g(\boldsymbol{x}, l). \tag{11}$$

Here, L_g is the number of levels in the 3D grid $G_\bullet$ and triplanes $\{T^1_\bullet, T^2_\bullet, T^3_\bullet\}$, $g(\boldsymbol{x}, l)$ is the (interpolated) feature vector at $\boldsymbol{x}$ for level l, w_l is a per-level dampening weight for anti-aliasing (Eq. 8) and '$\bigoplus$' is concatenation. We encode the direction $\boldsymbol{d}$ through spherical harmonics, $\text{SH}(\boldsymbol{d})$, as an auxiliary input to the color MLP (Eq. 9) that is independent of the feature vector $\Gamma_{\text{col}}(\boldsymbol{x})$.

We use low-resolution 3D grids and high-resolution triplanes, as in previous work (MERF [29]) to obtain the best rendering quality (Tab. 6). We double the grid resolution between levels and therefore have a low-resolution 3D grid with 3 levels (128^3–512^3) and higher-resolution triplanes with 7 levels (128^3–8192^3). Naïvely computing Eq. 11 requires $3+3\times7 = 24$ *texture* fetches (we rely on the CUDA texture API for hardware interpolation and do not need to explicitly query voxel corners/texels). As a render-time optimization, we save pre-summed features g'_L for each resolution level L (where we store $g'_L(\boldsymbol{v}) = \sum_{l=0}^{L} g(\boldsymbol{v}, l)$ for each texel $\boldsymbol{v}$ in L), such that Eq. 11 can be rewritten as $\Gamma(\boldsymbol{x}) = \bigoplus_{g \in \{G, T^1, T^2, T^3\}} [w_L g'_L(\boldsymbol{x}) + (1 - w_L) g'_{L-1}(\boldsymbol{x})]$ for $L = L(\boldsymbol{x})$ (Eq. 7). Here, $\boldsymbol{v}$ refers to the texel (voxel). Querying two levels requires only $2 + 3 \times 2 = 8$ texture fetches.

Table 6. **Grid feature layout.** We measure the effect of using only 3D or triplane features on the Eyeful Tower dataset [38], and note a significant drop in quality when compared to using both.

	Pinhole			Fisheye			Overall		
	↑PSNR	↑SSIM	↓LPIPS	↑PSNR	↑SSIM	↓LPIPS	↑PSNR	↑SSIM	↓LPIPS
3D Only	27.10	0.832	0.410	32.17	0.928	0.187	29.41	0.875	0.308
Triplane Only	<u>28.24</u>	<u>0.843</u>	<u>0.312</u>	<u>33.16</u>	<u>0.938</u>	<u>0.150</u>	<u>30.47</u>	<u>0.886</u>	<u>0.238</u>
Both	**29.07**	**0.880**	**0.268**	**34.57**	**0.952**	**0.115**	**31.57**	**0.913**	**0.198**

Table 7. **Depth error on ScanNet++ [41].** Our distance-adjusted Eikonal loss degrades geometric reconstruction less than other alternatives used to render unbounded scenes.

Method	↓Distance (m)	↓Distance (%)
Uniform Eikonal loss (world space)	**0.219**	**8.56**
Uniform Eikonal loss (contracted space)	0.419	16.11
No Eikonal loss	0.996	29.93
Distance-adjusted Eikonal loss (ours)	<u>0.221</u>	<u>11.13</u>

D. Geometric Reconstruction

We evaluate geometric reconstruction on ScanNet++ [41] (which has "ground-truth" laser scan depth only for foreground pixels) in Tab. 7 for the strategies in Fig. 6. Using uniform Eikonal loss in contracted space degrades accuracy (0.419 m error vs 0.219 m for uniform world space and 0.221 m with our distance-adjusted method) and omitting Eikonal loss gives the worst results (0.996 m).

E. ScanNet++

We evaluate 9 scenes from ScanNet++ [41] in Sec. 4.3 (5FB5D2DBF2, 8B5CAF3398, 39F36DA05B, 41B00FEDDB, 56A0EC536C, 98B4EC142F, B20A261FDF, F8F12E4E6B, FE1733741F). We undistort the fisheye DSLR captures to pinhole images using the official dataset toolkit [42] to facilitate comparisons against 3D Gaussian splatting [14] (whose implementation does not support fisheye projection). We use the official validation splits, which consist of entirely novel trajectories that present a more challenging novel-view synthesis problem than the commonly used pattern of holding out every eighth frame [21]. The dataset authors note that their release is still in the beta testing phase, and that the final layout is subject to change. Our testing reflects the dataset as of November 2023.

F. Societal Impact

Our technique facilitates the rapid generation of high-quality neural representations. Consequently, the risks associated with our work parallel those found in other neural rendering studies, primarily centered around privacy and security issues linked to the deliberate or unintentional capture of sensitive information, such as human facial features and vehicle license plate numbers. Although we did not specifically apply our approach to data involving privacy or security concerns, there exists a risk, akin to other neural rendering methodologies, that such sensitive data could become incorporated into the trained model if the datasets utilized are not adequately filtered beforehand. It is imperative to engage in pre-processing of the input data employed for model training, especially when extending its application beyond research, to ensure the model's resilience against privacy issues and potential misuse. However, a more in-depth exploration of this matter is beyond the scope of this paper.