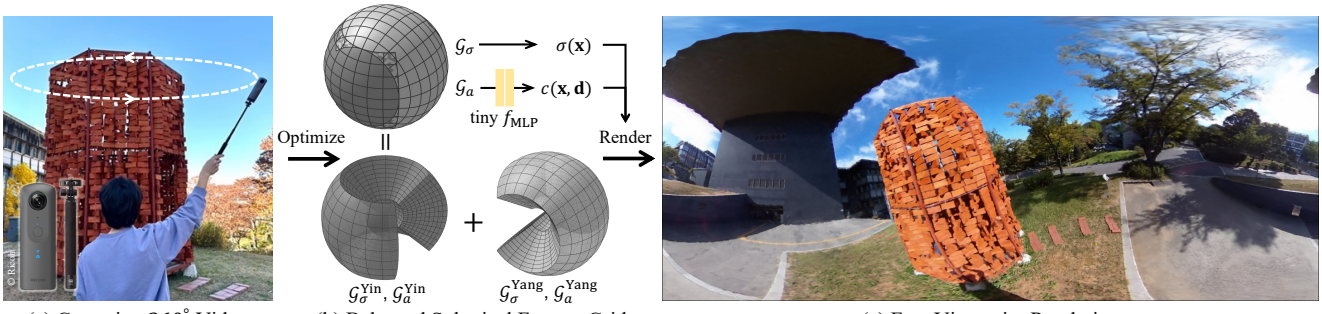
Balanced Spherical Grid for Egocentric View Synthesis

Changwoon Choi[1], Sang Min Kim[1], Young Min Kim[1,2]
[1]Dept. of Electrical and Computer Engineering, Seoul National University, Korea
[2]Interdisciplinary Program in Artificial Intelligence and INMC, Seoul National University



(a) Capturing 360° Video (b) Balanced Spherical Feature Grid (c) Free Viewpoint Rendering

Figure 1. We propose a practical solution to reconstruct large-scale scenes from a short egocentric video. (a) Our scalable capturing setup observes the holistic environment by casually swiping a selfie stick with an omnidirectional camera attached. (b) Then we optimize our balanced spherical feature grids which are tailored for the outward-looking setup. (c) EgoNeRF can quickly train and render high-quality images at nearby positions. Project page: https://changwoon.info/publications/EgoNeRF

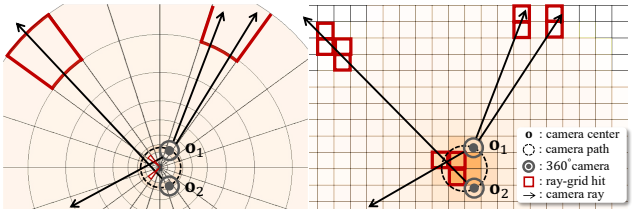
Abstract

We present EgoNeRF, a practical solution to reconstruct large-scale real-world environments for VR assets. Given a few seconds of casually captured 360 video, EgoNeRF can efficiently build neural radiance fields. Motivated by the recent acceleration of NeRF using feature grids, we adopt spherical coordinate instead of conventional Cartesian coordinate. Cartesian feature grid is inefficient to represent large-scale unbounded scenes because it has a spatially uniform resolution, regardless of distance from viewers. The spherical parameterization better aligns with the rays of egocentric images, and yet enables factorization for performance enhancement. However, the naïve spherical grid suffers from singularities at two poles, and also cannot represent unbounded scenes. To avoid singularities near poles, we combine two balanced grids, which results in a quasi-uniform angular grid. We also partition the radial grid exponentially and place an environment map at infinity to represent unbounded scenes. Furthermore, with our resampling technique for grid-based methods, we can increase the number of valid samples to train NeRF volume. We extensively evaluate our method in our newly introduced synthetic and real-world egocentric 360 video datasets, and it consistently achieves state-of-the-art performance.

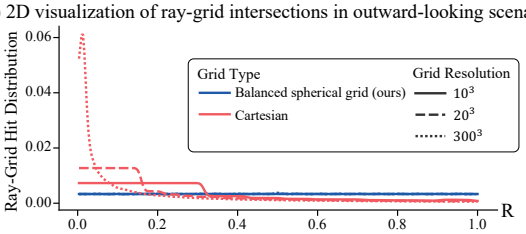
1. Introduction

With the recent advance in VR technology, there exists an increasing need to create immersive virtual environments. While a synthetic environment can be created by expert designers, various applications also require transferring a real-world environment. Spherical light fields [4–6,26,28] can visualize photorealistic rendering of the real-world environment with the help of dedicated hardware with carefully calibrated multiple cameras. A few works [3, 16] also attempt to synthesize novel view images by reconstructing an explicit mesh from an egocentric omnidirectional video. However, their methods consist of complicated multi-stage pipelines and require pretraining for optical flow and depth estimation networks.

In this paper, we build a system that can visualize a large-scale scene without sophisticated hardware or neural networks trained with general scenes. We utilize panoramic images, as suggested in spherical light fields. However, we acquire input with a commodity omnidirectional camera with two fish-eye lenses instead of dedicated hardware. As shown in Fig. 1 (a), the environment can be captured with the omnidirectional camera attached to a selfie stick within less than five seconds. Then the collected images observe a large-scale scene that surrounds the viewpoints. We introduce new synthetic and real-world datasets of omnidirec-



(a) 2D visualization of ray-grid intersections in outward-looking scenario



(b) Ray-grid intersection distribution along distance from center

Figure 2. (a) When the camera trajectory is short relative to the scene size, the proposed balanced spherical grid (left) exhibits uniform hitting rate for grid cells whereas the conventional Cartesian grid (right) suffers from non-uniform ray-grid hits. The orange shade indicates the relative density of hit count of the grid cells. (b) Experiments show that spherical coordinates achieve nearly uniform ray-grid hit distribution, while Cartesian coordinate is biased to the center especially when we use a fine-resolution grid.

tional videos acquired from both indoor and outdoor scenes. Combined with Neural Radiance Fields (NeRF) [22], the images can train a neural volume that can render fine details or view-dependent effects without explicit 3D models.

To this end, we present Egocentric Neural Radiance Fields, or EgoNeRF, which is the neural volume representation tailored to egocentric omnidirectional visual input. Although NeRF and its variants with MLP-based methods show remarkable performance in view synthesis, they suffer from lengthy training and rendering time. The recent Cartesian feature grids can lead to faster convergence [7, 31] for rendering a bounded scene with an isolated object, but they have several limitations for our datasets which mostly contain inside-out views of large scenes: (1) The uniform grid size, regardless of distance from the camera, is insufficient to represent fine details of near objects and extravagant for coarse integrated information from far objects. (2) Cartesian grid suffers from non-uniform ray-grid hits in the egocentric scenario as demonstrated in Fig. 2, thus, as pointed in [31], prior arts need careful training strategies such as progressive scaling [7, 31] or view-count-adaptive per-voxel learning rate [31]. EgoNeRF models the volume using a spherical coordinate system to cope with the aforementioned limitations. Figure 3 shows that EgoNeRF converges faster compared to MLP-based methods (NeRF [22] and mip-NeRF 360 [2]) and has higher performance compared to Cartesian grid methods (TensoRF [7] and DVGO [31]).

Our spherical grid is designed to be balanced in any direction, which leads to a more efficient data structure for the large-scale environment. The naïve spherical grid contains high valence vertices at two poles, and, when adapted as a feature grid for neural volume rendering, the polar regions suffer from undesirable artifacts. We exploit a quasi-uniform angular grid by combining two spherical grids [18]. In the radial direction, the grid intervals increase exponentially, which not only allows our representation to cover large spaces but also makes the spherical frustum have a similar length in the angular and radial directions. We add an environment map at infinite depth, which is especially useful for outdoor environments with distant backgrounds such as skies. Last but not least, we propose an efficient hierarchical sampling method exploiting our density feature grid without maintaining an additional coarse density grid.

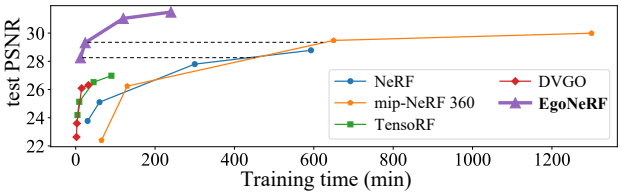


Figure 3. Training curve comparison in *OmniBlender* scenes.

We demonstrate that our proposed approach can lead to faster convergence and high-quality rendering with a small memory footprint in various scenarios for large-scale environments. EgoNeRF is expected to create a virtual rendering of large scenes from data captured by non-expert users, which cannot be easily modeled with 3D assets or conventional NeRF.

2. Related Works

Visualizing Omnidirectional View of Scenes Panoramic images are widely used in many applications for remote experiences. After captured by photo-stitching apps or dedicated hardware, they allow users to rotate around the captured position. However, we need additional information to allow the full 6 DoF movement in the scene. Prior works propose sophisticated camera rigs to capture spherical light fields [4–6, 26, 28]. Given multi-view images, they enable synthesizing images at novel viewpoints by reconstructing 3D mesh or multi-sphere images instead of multi-plane images in ordinary images. With additional depth information, recent works demonstrate novel view synthesis with a single panoramic image [12, 14]. The depth channel is acquired from RGBD camera or approximated coarse planar facades.

In contrast, we assume more casual input, using commodity 360° camera with two fish-eye lenses to capture a short video clip of the large-scale scene. A few works also explored the same setup [3, 16] and represented the scene with a deformed proxy mesh with texture maps us-

ing pre-trained neural networks for optical flow and depth estimation. Our pipeline is simpler as we train a neural network with the captured sequence of images without any pre-trained network. We combine the visualization pipeline for large-scale scenes with NeRF formulation and can capture complex view-dependent effects and fine structures, unlike reconstructed textured mesh.

Practical Variants of NeRF NeRF [22] flourished in the field of novel view synthesis, showing photorealistic quality with its simple formulation. However, the original NeRF formulation exhibits clear drawbacks, such as lengthy training and rendering time, and the difficulty of deformation or scene edits. Many follow-up works exploded, overcoming the limitations in various aspects [1,2,8,21,25,27,30]. Here we specifically focus on practical extension for fast rendering and training. NeRF represents a scene as a single MLP that maps coordinates into color and volume density. It is slow in rendering and optimization as the volume rendering requires multiple forward passes of the MLP.

To accelerate the rendering speed, radiance is represented with an explicit voxel grid storing features [13, 20, 35]. However, they train the network by distilling information from pre-trained NeRF, which even lengthens the training time. More recent works exploit various data structures to directly optimize the feature grid [7, 10, 24, 31]. They have shown that employing an explicit feature grid achieves fast optimization without sacrificing quality. The feature grids are defined on the Cartesian coordinate system, which assumes a scene within a bounding box. These are not suitable for representing large-scale scenes whose viewpoints observe outside of the captured locations.

The naïve strategy to choose ray samples wastes most samples and it leads to slow convergence since many regions are either free spaces or occluded by other objects in the real world. To increase the sample efficiency, the original NeRF [22] employs a hierarchical sampling strategy for the volume density and maintains two density MLPs for coarse and fine resolution, respectively. In the same context, Müller et al. [24] maintain additional multi-scale occupancy grids to skip ray marching steps. Hu et al. [15] allocate dense momentum voxels for valid sampling, and Sun et al. [31] also use an extra coarse density voxel grid. Maintaining separate coarse feature grids or neural networks requires additional memory and increases computational burdens. We propose an efficient sampling strategy and quickly train a volume that represents a large-scale environment.

3. Feature Grid Representation for EgoNeRF

EgoNeRF utilizes feature grids to accelerate the neural volume rendering of NeRF. Feature grids in previous works employ a Cartesian coordinate system, which regularly partition the volume in xyz axis [13, 20, 35]. To better express the egocentric views captured from omnidirectional videos, we use a spherical coordinate system. We modify the spherical coordinate in both angular and radial partitions to efficiently express outward views of the surrounding environment, as described in Sec. 3.1. For rendering and training, the values are interpolated from the feature grid, which can be further factorized to reduce the memory and accelerate the learning [7] (Sec. 3.2). With our balanced feature grid, individual cells produce a uniform hitting rate of rays.

3.1. Balanced Spherical Grid

Our balanced spherical grid is composed of the angular partition and the radial partition.

Angular Partitions The desirable angular partition should result in regular shapes and be easily parameterized. When we regularly partition on the angle parameters, the naïve spherical coordinate system results in irregular grid partitions, which severely distort the two polar regions. Existing regular partitions do not maintain orthogonal axis parameterization [11], which hinders further factorization.

As a simple resolution, we only use the quasi-uniform half of the ordinary spherical coordinate system and combine two of them [18]. The two grids are referred to as the Yin grid and Yang grid, respectively, which have identical shapes and sizes as shown in Fig. 1 (b) and Fig. 4 (a). Together they can cover the entire sphere with minimal overlap, similar to the two regions of a tennis ball.

The Yin grid is defined as:

$$(\pi/4 \leq \theta \leq 3\pi/4) \cap (-3\pi/4 \leq \phi \leq 3\pi/4), \quad (1)$$

where θ is colatitude and ϕ is longitude. The axis of another component grid, namely the Yang grid, is located at the equator of the Yin grid:

$$\begin{bmatrix} x^{\text{Yin}} \\ y^{\text{Yin}} \\ z^{\text{Yin}} \end{bmatrix} = M \begin{bmatrix} x^{\text{Yang}} \\ y^{\text{Yang}} \\ z^{\text{Yang}} \end{bmatrix}, M = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}. \quad (2)$$

We discretize the angular grid of Yin and Yang grid by N_θ^y and N_ϕ^y partitions for θ^y, ϕ^y axis respectively, where $y \in \{\text{Yin}, \text{Yang}\}$. The partition is uniform in angles leading to the grid size of

$$\Delta\theta^y = \frac{\pi}{2}\frac{1}{N_\theta^y}, \; \Delta\phi^y = \frac{3\pi}{2}\frac{1}{N_\phi^y}. \quad (3)$$

Radial Partitions By adopting the spherical coordinate system, the grid cells cover larger regions as r increases. This is desired in the egocentric setup, as the panoramic image capture more detailed close-by views of central objects while distant objects occupy a small area on the projected images. We further make the grid along the r axis increase

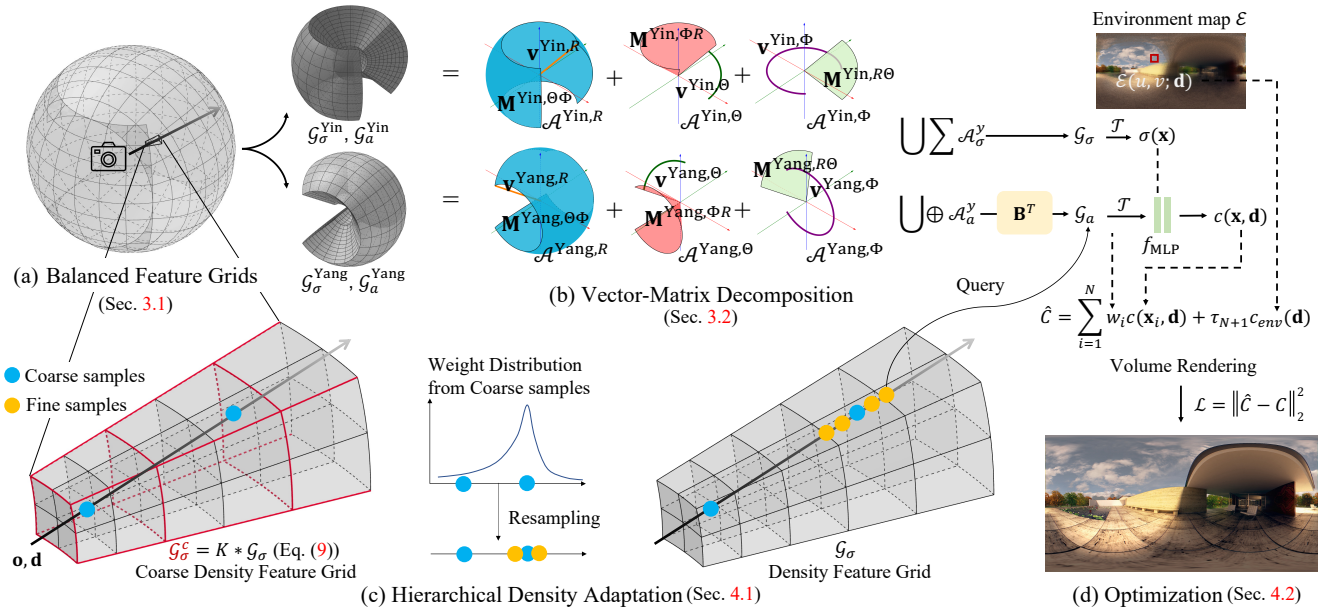


Figure 4. Overview of our method. (a) We represent radiance fields as features stored in balanced feature grids $\mathcal{G}_\sigma, \mathcal{G}_a$, (b) which are further decomposed into vector and matrix components. (c) The hierarchical sampling is conducted by obtaining a coarse density grid from the density feature grid on the fly during optimization. (d) The balanced feature grids are optimized with photometric loss.

exponentially for far regions such that the resulting cell exhibit similar lengths in the angular and radial direction.

Specifically, if we denote the radial scales of both the Yin and Yang grids as r^y,

$$r_i^y = r_0 k^{i-1}, \; R_{\text{max}} = r_0 k^{N_r^y - 1}, \tag{4}$$

where R_{max} is the radius of the scene bounding sphere and constant value r_0 is the radius of the first spherical shell. We set the grid interval to r_0 for the grid interval less than r_0.

We can optionally use the environment map for outdoor or large indoor environments. Our spherical grid is still bounded by R_{max}, limiting the size of the environment. The environment map denoted as $\mathcal{E} \in \mathbb{R}^{H \times W \times 3}$, is a simple equirectangular image and represents what is visible at an almost infinite distance.

3.2. Feature Grid as Radiance Field

Now we describe our radiance field representation with the balanced spherical feature grid. Given a set of egocentric images with corresponding camera parameters, EgoNeRF aims to reconstruct 3D scene representation and synthesize novel view images. Instead of regressing for the volume density σ and color c from MLP [22], we build explicit feature grids of the density $\mathcal{G}_\sigma$ and the appearance $\mathcal{G}_a$ which serve as the mapping function. Both grids are composed of our balanced spherical grids of resolution $2N_r^y \times N_\theta^y \times N_\phi^y$, as defined in Sec. 3.1. The density grid $\mathcal{G}_\sigma \in \mathbb{R}^{2N_r^y \times N_\theta^y \times N_\phi^y}$ has a single channel which stores the explicit volume density value, and the appearance grid $\mathcal{G}_a \in \mathbb{R}^{2N_r^y \times N_\theta^y \times N_\phi^y \times C}$ stores C-dimensional neural appearance features. The volume density and color at position $\mathbf{x}$ and viewing direction $\mathbf{d}$ are obtained by

$$\sigma(\mathbf{x}) = \mathcal{T}(\mathcal{G}_\sigma, \mathbf{x}), \; c(\mathbf{x}, \mathbf{d}) = f_{\text{MLP}}(\mathcal{T}(\mathcal{G}_a, \mathbf{x}), \mathbf{d}), \tag{5}$$

where $\mathcal{T}$ denotes a trilinear interpolation, and f_{MLP} is a tiny MLP that decodes the neural feature to color.

Inspired by [7], we further decompose the feature tensor into vectors and matrices as shown in Fig. 4 (b):

$$\begin{aligned}\mathcal{G}_\sigma^y &= \sum_{n=1}^{N_\sigma} \mathbf{v}_{\sigma,n}^{y,R} \otimes \mathbf{M}_{\sigma,n}^{y,\Theta\Phi} + \mathbf{v}_{\sigma,n}^{y,\Theta} \otimes \mathbf{M}_{\sigma,n}^{y,\Phi R} + \mathbf{v}_{\sigma,n}^{y,\Phi} \otimes \mathbf{M}_{\sigma,n}^{y,R\Theta} \\ &= \sum_{n=1}^{N_\sigma} \sum_{m \in R\Theta\Phi} \mathcal{A}_{\sigma,n}^{y,m}, \end{aligned} \tag{6}$$

$$\mathcal{G}_a^y = \sum_{n=1}^{N_a} \mathcal{A}_{a,n}^{y,R} \otimes \mathbf{b}_{3n-2}^y + \mathcal{A}_{a,n}^{y,\Theta} \otimes \mathbf{b}_{3n-1}^y + \mathcal{A}_{a,n}^{y,\Phi} \otimes \mathbf{b}_{3n}^y, \tag{7}$$

$$\mathcal{G}_\sigma = \bigcup_{y \in Y} \mathcal{G}_\sigma^y, \mathcal{G}_a = \bigcup_{y \in Y} \mathcal{G}_a^y, Y = \{\text{Yin, Yang}\}, \tag{8}$$

where $\otimes$ represents the outer product and $\mathbf{v}, \mathbf{b}, \mathbf{M}$ represents vector and matrix factors. This low-rank tensor factorization significantly reduces the space complexity from $\mathcal{O}(n^3)$ to $\mathcal{O}(n^2)$. With the minimal overhead of storing two grids, we can maintain regular angular components and yet factorize the grid using spherical parameterization. The full

decomposed formulation is described in the supplementary material.

4. Training EgoNeRF

We utilize the balanced spherical grids to represent the volume density σ and color c, which are stored in $\mathcal{G}_\sigma$ and $\mathcal{G}_a$, respectively. In this chapter, we describe the technical details of the optimization process of our proposed method.

4.1. Hierarchical Density Adaptation

As the scenes typically contain sparse occupied regions, we adapt the hierarchical sampling strategy of the original NeRF [22] for feature grids. While other recent variants using feature grid [15, 24, 31] maintain a dedicated data structure for the coarse grid, we exploit our dense geometry feature grid $\mathcal{G}_\sigma$ for the first coarse sampling stage without allocating additional memory for the coarse grid.

The hierarchical sampling strategy first samples coarse N_c points along the ray to obtain a density estimate σ from which we can sample fine N_f points with importance sampling. However, evaluating σ with dense $\mathcal{G}_\sigma$ at the coarsely sampled points might skip the important surface regions. Therefore, we obtain σ value from a coarser density feature grid which can be obtained on the fly by applying a non-learnable convolution kernel K:

$$\sigma(\mathbf{x}_{\text{coarse}}) = \mathcal{T}(\mathcal{G}^c_\sigma, \mathbf{x}_{\text{coarse}}) = \mathcal{T}(K * \mathcal{G}_\sigma, \mathbf{x}_{\text{coarse}}). \quad (9)$$

We use the average pooling kernel as K. It is reasonable to define a coarse grid by convolving the dense grid because our density grid $\mathcal{G}_\sigma$ stores the volume density itself, which has a physical meaning, not neural features.

From the volume density values of coarsely sampled points, we calculate weights for importance sampling by

$$w_i = \tau_i(1 - e^{-\sigma_i \delta_i}), i \in [1, N_c], \quad (10)$$

where δ_i is the distance between coarse samples, $\tau_i = e^{-\sum_{j=1}^{i-1} \sigma_j \delta_j}$ is the accumulated transmittance. Then the fine N_f locations are sampled from the filtered probability distribution. Finally, the volume density σ and color c at $N_c + N_f$ samples are used to render pixels.

4.2. Optimization

The images of EgoNeRF are synthesized by applying the volume rendering equation along the camera ray [22] and the optional environment map. Specifically, the points $\mathbf{x}_i = \mathbf{o} + t_i\mathbf{d}$ along the camera ray from camera position $\mathbf{o}$ and ray direction $\mathbf{d}$ are accumulated to find the pixel value by

$$\hat{C} = \sum_{i=1}^{N} \tau_i(1 - e^{-\sigma(\mathbf{x}_i)\delta_i})c(\mathbf{x}_i, \mathbf{d}) + \tau_{N+1} c_{\text{env}}(\mathbf{d}). \quad (11)$$

$N = N_c + N_f$ is the number of samples as described in Sec. 4.1. $\sigma(\mathbf{x})$ and $c(\mathbf{x}, \mathbf{d})$ are obtained from our balanced feature grids in Eq. (5). Since the size of our feature grid is exponentially increasing along the r direction, we distribute N_c coarse samples exponentially rather than uniformly. The second term in Eq. (11) is fetched from the environment map

$$c_{\text{env}}(\mathbf{d}) = \mathcal{E}(u, v; \mathbf{d}), \quad (12)$$

where the sampling position (u, v) is only dependent on the viewing direction $\mathbf{d}$. The effect of the environment map is further discussed in Sec. 5.3.

Finally, we optimize the photometric loss between rendered images and training images

$$\mathcal{L} = \frac{1}{|\mathcal{R}|} \sum_{\mathbf{r} \in \mathcal{R}} \left\| \hat{C}(\mathbf{r}) - C(\mathbf{r}) \right\|_2^2, \quad (13)$$

where $\mathcal{R}$ is a randomly sampled ray batch, $\hat{C}(\mathbf{r}), C(\mathbf{r})$ are rendered and the ground-truth color of the pixel corresponding to ray $\mathbf{r}$. With the simple photometric loss, our feature grids $\mathcal{G}_\sigma, \mathcal{G}_a$, decoding MLP f_{MLP}, and environment map $\mathcal{E}$ are jointly optimized. For real-world datasets, in which camera poses are not perfect, we additionally optimize a TV loss [29] at our feature grid to reduce noise. Furthermore, since our balanced feature grid guarantees a nearly uniform ray-grid hitting rate, EgoNeRF does not need a coarse-to-fine reconstruction approach for robust optimization used in other feature grid-based methods [7, 31].

5. Experiments

We demonstrate that EgoNeRF can quickly capture and synthesize novel views of large-scale scenes. We describe full implementation details including hyperparameter setup in the supplementary material.

Datasets Since many of the existing datasets for NeRF are dedicated to a setup where a bounded object is captured from outside-in viewpoints, we propose new synthetic and real datasets of large-scale environments captured with omnidirectional videos. *OmniBlender* is a realistic synthetic dataset of 11 large-scale scenes with detailed textures and sophisticated geometries in both indoor and outdoor environments, 25 images for both train and test, respectively. It consists of omnidirectional images along a relatively small circular camera path. The spherical images are rendered using Blender's Cycles path tracing renderer [9] with 2000×1000 resolution. *Ricoh360* is a real-world 360° video dataset captured with a Ricoh Theta V camera with 1920×960 resolution. We record video on the circular path by rotating an omnidirectional camera fixed with a selfie stick as shown in Fig. 1 (a). The dataset consists of 11 diverse indoor and outdoor scenes, 50 images for train and

Step	Method	*OmniBlender* Indoor PSNR	Indoor $PSNR^{WS}$	Indoor LPIPS	Indoor SSIM	Indoor $SSIM^{WS}$	*OmniBlender* Outdoor PSNR	Outdoor $PSNR^{WS}$	Outdoor LPIPS	Outdoor SSIM	Outdoor $SSIM^{WS}$	*Ricoh360* PSNR	*Ricoh360* $PSNR^{WS}$	*Ricoh360* LPIPS	*Ricoh360* SSIM	*Ricoh360* $SSIM^{WS}$
5k	NeRF [22]	26.25	27.27	0.500	0.726	0.710	22.36	23.62	0.524	0.651	0.611	22.09	23.82	0.576	0.649	0.623
	mip-NeRF 360 [2]	23.51	24.41	0.628	0.649	0.613	21.76	23.03	0.545	0.614	0.568	22.30	24.12	0.555	0.632	0.604
	TensoRF [7]	25.91	26.93	0.553	0.722	0.708	23.21	24.74	0.500	0.672	0.645	23.20	25.16	0.542	0.676	0.658
	DVGO [31]	24.26	25.29	0.633	0.689	0.666	21.70	23.15	0.570	0.642	0.605	22.45	24.59	0.573	0.664	0.646
	EgoNeRF	28.87	30.06	0.310	0.803	0.803	27.90	29.31	0.167	0.844	0.832	24.52	26.74	0.331	0.737	0.729
10k	NeRF [22]	27.66	28.80	0.425	0.756	0.749	23.63	24.90	0.458	0.686	0.650	22.78	24.49	0.538	0.663	0.638
	mip-NeRF 360 [2]	27.41	28.47	0.412	0.763	0.755	25.57	26.80	0.306	0.769	0.741	24.28	26.28	0.384	0.725	0.710
	TensoRF [7]	26.96	26.98	0.469	0.751	0.743	24.09	25.71	0.436	0.696	0.676	23.82	25.75	0.481	0.694	0.678
	DVGO [31]	25.44	26.53	0.556	0.715	0.699	22.54	24.06	0.518	0.659	0.628	23.08	25.28	0.529	0.678	0.664
	EgoNeRF	30.23	31.47	0.248	0.840	0.841	28.81	30.21	0.136	0.868	0.859	24.71	26.98	0.314	0.746	0.740
100k	NeRF [22]	31.67	33.08	0.240	0.852	0.853	27.12	28.54	0.269	0.789	0.772	24.91	26.65	0.384	0.721	0.702
	mip-NeRF 360 [2]	31.12	32.41	0.225	0.859	0.859	29.34	30.63	0.135	0.879	0.867	25.57	27.62	0.268	0.778	0.770
	TensoRF [7]	29.25	30.57	0.376	0.791	0.793	25.68	27.47	0.344	0.734	0.726	25.16	27.13	0.376	0.732	0.724
	DVGO [31]	28.84	30.23	0.348	0.798	0.803	24.87	26.73	0.363	0.720	0.711	24.90	27.28	0.376	0.732	0.729
	EgoNeRF	33.11	34.53	0.142	0.902	0.904	30.56	32.04	0.087	0.904	0.901	25.25	27.50	0.286	0.763	0.758

Table 1. Quantitative results in outward-looking *OmniBlender* and *Ricoh360* dataset. Top results are colored as top1 , top2 , and top3 .

test, respectively. With the benefit of the simple procedure, the whole data acquisition is finished in less than 5 seconds, which enables capturing the surrounding scene while it remains nearly static. We obtain camera poses using SfM library OpenMVG [23]. A detailed description of our dataset can be found in the supplementary material.

Baselines EgoNeRF is a variant of NeRF [22], which synthesizes novel views of the scene using the neural volume trained with multi-view images. However, the original NeRF utilizes an MLP to represent the scene volume. There also exists a recent variant called mip-NeRF 360 [2], which combines many techniques to increase the quality of the results, including the adaptation to unbounded scenes by warping space farther than a certain radius. EgoNeRF employs feature grids and vector-matrix factorization in the balanced spherical grid. DVGO [31] exploits feature grids in a Cartesian coordinate with great acceleration, whereas TensoRF [7] deploys factorization, also in Cartesian coordinate. For all the methods, we train with the same ray batch size and the same number of feature grids (for DVGO and TensoRF) with one RTX-3090 GPU for a fair comparison.

5.1. Quantitative Results

The quantitative results are reported in mean PSNR, SSIM [33], and LPIPS [36] across test images in *OmniBlender* and *Ricoh360* dataset in Tab. 1. Since equirectangular images in our datasets have distortion near poles, we additionally measure weighted-to-spherically uniform PSNR and SSIM [32] ($PSNR^{WS}$ and $SSIM^{WS}$), which place smaller weights near the poles when evaluating the metrics.

Table 1 shows that EgoNeRF outperforms all compared methods across all error metrics in the *OmniBlender* dataset. With the efficient grid structure of EgoNeRF, the difference is more significant in earlier iterations. Considering the time for each iteration, the efficiency gap is even more significant, which is also visualized in Fig. 3. Our approach shows high performance even at the early 5k steps, which takes 10 minutes of wall-clock time. In contrast, mip-NeRF 360 needs approximately 8 hours to outperform our results at 5k steps. In *Ricoh360*, EgoNeRF surpasses other methods in 5k and 10k training steps, and shows comparable results in 100k steps. However, our approach sometimes produces spotty artifacts in real-world datasets because the camera pose estimates can be erroneous. On the other hand, MLP-based methods show blurry rendering, which sporadically achieves better scores in error metrics. Such a phenomenon is prominent when the error in the camera pose makes the rays hit neighboring cells in the feature grid, which is further discussed in the supplementary material.

More importantly, the feature grid using the Cartesian coordinate system (TensoRF and DVGO) results in inferior performance, especially in outdoor scenes. This supports our main claim that the Cartesian grid is inadequate to represent large-scale scenes captured from egocentric viewpoints. In contrast, MLP-based methods (NeRF and mip-NeRF 360) achieve moderate results.

5.2. Qualitative Results

The qualitative results in *OmniBlender* and *Ricoh360* datasets are demonstrated in Fig. 5. Our method reconstructs fine details in both close-by objects and far-away regions. However, for Cartesian grid-based methods (TensoRF and DVGO), many cells are wasted without being trained in far objects, while center cells might not have a sufficient resolution as depicted in Fig. 2. It results in visible artifacts in the picture in *BarberShop*, bike in *BistroBike*, bricks in *bricks*, and posters in *poster*. This phenomenon is predominant in large scenes, while EgoNeRF gives consistently faithful results regardless of the size of the scenes.

MLP-based approaches show better visual results than Cartesian feature grid-based methods with much longer

(a) *OmniBlender* (b) *Ricoh360*

Figure 5. Comparative results of novel view synthesis on the outward-looking (a) synthetic *OmniBlender* dataset and (b) real-world *Ricoh360* dataset. Best viewed on screen.

Method	Indoor			Outdoor		
	PSNR	LPIPS	SSIM	PSNR	LPIPS	SSIM
w/o exp R grid	31.32	0.188	0.871	26.66	0.187	0.792
w/o Yin-Yang grid	30.53	0.191	0.860	26.74	0.160	0.806
Spherical Grid	30.78	0.209	0.858	26.25	0.213	0.773
w/o Resampling	32.40	0.167	0.886	30.12	0.105	0.891
w/o Environment map		-		30.04	0.107	0.891
EgoNeRF (full)	33.11	0.142	0.902	30.56	0.087	0.904

Table 2. An ablation study in *OmniBlender* dataset. We replace and remove important components in EgoNeRF.

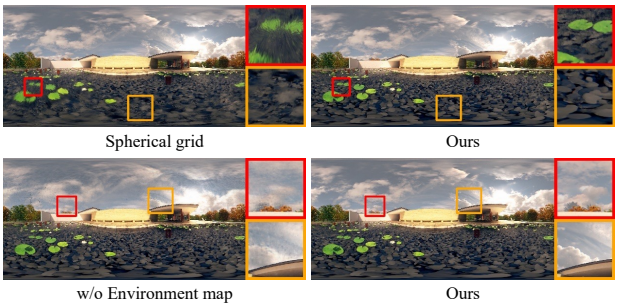


Figure 6. Qualitative results of ablation study.

training and rendering time. However, mip-NeRF 360 often misses fine structures: e.g., stick of broom, thin handle and support fixture in tray, headrest attachment in chair in *Barbershop*, street lamp, side mirror of bike, small colorful lightbulbs and thin wire in *BistroBike*. Some of them are also indicated with white dotted circles in Fig. 5. This may be due to mip-NeRF 360 resample ray samples from the proposal MLP and do not apply rendering loss for the proposal MLP to relieve lengthy training time to optimize large MLPs, thus the weight distribution is strongly determined by the initial guess of the proposal MLP. In contrast, since our approach shares the same density grid $\mathcal{G}_\sigma$ to query volume density at coarse samples and fine samples, EgoNeRF shows superior rendering results on fine details. Also, MLP-based approaches show smoothed rendering results across all the scenes (e.g., windows are blurred, cannot see the sky through the gap between bricks, the boundaries between stepping blocks are blurred in *Bricks*, two reflected lights are merged in *poster*. Some of them are also highlighted with white dotted circles in Fig. 5.), while EgoNeRF shows high-quality images similar to ground-truth images.

5.3. Ablation study

We analyze the effects of important components of EgoNeRF with ablated versions. Table 2 shows that removing any of the components in our model degrades the performance across all metrics. The first three rows are related to the balanced spherical grid. Using the uniform radial partition deteriorates the performance, especially in outdoor scenes. Without Yin-Yang grids, the angular partition exhibits high valence grid points on two poles and degrades the error metrics consequently. Removing both radial and angular balanced grids, which is identical to uniform spherical grids, causes the biggest drop in performance except PSNR in indoor scenes. As shown in the first row of Fig. 6, the spherical feature grid has radial direction artifacts (red box) and shows blurrier rendered results for nearby objects compared to our full model. Also, not employing resampling techniques and using a double number of ray samples reduces performance. Lastly, removing the environment map in outdoor scenes shows blurry artifacts in infinitely far regions as shown in the second row of Fig. 6 and reduces the performance consequently.

We provide additional analysis on the impact of hyperparameters, scene depths, and out-of-distribution testing in the supplementary material.

6. Conclusion

We present EgoNeRF, an efficient adaptation of the NeRF into large-scale scenes with casual input. We utilize a balanced spherical feature grid and maintain uniform ray hit rates for individual cells for scenes captured with a short video of omnidirectional cameras. Together with factorization and resampling techniques, we can achieve fast and high-quality rendering of various indoor and outdoor environments.

Although EgoNeRF significantly outperforms the prior works in terms of visual quality and our approach converges much faster than MLP-based methods, we have some limitations. In this paper, we do not consider all the challenges that come from real-world scenarios such as photometric variation from automatic camera exposure. EgoNeRF sometimes shows noisy artifacts when the camera poses are not correct in the real-world *Ricoh360* dataset, while MLP-based algorithms output blurred images. Further analysis of the impact of camera parameter error is provided in the supplementary material. One can resolve this by jointly optimizing the camera parameters as in [17, 19, 34]. Furthermore, like other NeRF-based models, we assume that scenes are static.

Acknowledgements This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2023-00208197) and Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) [NO.2021-0-01343, Artificial Intelligence Graduate School Program (Seoul National University)]. Young Min Kim is the corresponding author.

References

[1] Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5855–5864, October 2021. 3

[2] 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 (CVPR)*, pages 5470–5479, June 2022. 2, 3, 6, 7

[3] Tobias Bertel, Mingze Yuan, Reuben Lindroos, and Christian Richardt. Omniphotos: casual 360 vr photography. *ACM Transactions on Graphics (TOG)*, 39(6):1–12, 2020. 1, 2

[4] Michael Broxton, Jay Busch, Jason Dourgarian, Matthew DuVall, Daniel Erickson, Dan Evangelakos, John Flynn, Peter Hedman, Ryan Overbeck, Matt Whalen, et al. Deepview immersive light field video. In *ACM SIGGRAPH 2020 Immersive Pavilion*, pages 1–2. 2020. 1, 2

[5] Michael Broxton, Jay Busch, Jason Dourgarian, Matthew DuVall, Daniel Erickson, Dan Evangelakos, John Flynn, Ryan Overbeck, Matt Whalen, and Paul Debevec. A low cost multi-camera array for panoramic light field video capture. In *SIGGRAPH Asia 2019 Posters*, pages 1–2. 2019. 1, 2

[6] Michael Broxton, John Flynn, Ryan Overbeck, Daniel Erickson, Peter Hedman, Matthew Duvall, Jason Dourgarian, Jay Busch, Matt Whalen, and Paul Debevec. Immersive light field video with a layered mesh representation. *ACM Transactions on Graphics (TOG)*, 39(4):86–1, 2020. 1, 2

[7] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European Conference on Computer Vision (ECCV)*, 2022. 2, 3, 4, 5, 6, 7

[8] Changwoon Choi, Juhyeon Kim, and Young Min Kim. Ibl-nerf: Image-based lighting formulation of neural radiance fields. *arXiv preprint arXiv:2210.08202*, 2022. 3

[9] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. 5

[10] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5501–5510, June 2022. 3

[11] Gene Greger, Peter Shirley, Philip M Hubbard, and Donald P Greenberg. The irradiance volume. *IEEE Computer Graphics and Applications*, 18(2):32–43, 1998. 3

[12] Takayuki Hara and Tatsuya Harada. Enhancement of novel view synthesis using omnidirectional image completion. *arXiv preprint arXiv:2203.09957*, 2022. 2

[13] Peter Hedman, Pratul P Srinivasan, Ben Mildenhall, Jonathan T Barron, and Paul Debevec. Baking neural radiance fields for real-time view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5875–5884, 2021. 3

[14] Ching-Yu Hsu, Cheng Sun, and Hwann-Tzong Chen. Moving in a 360 world: Synthesizing panoramic parallaxes from a single panorama. *arXiv preprint arXiv:2106.10859*, 2021. 2

[15] Tao Hu, Shu Liu, Yilun Chen, Tiancheng Shen, and Jiaya Jia. Efficientnerf efficient neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12902–12911, June 2022. 3, 5

[16] Hyeonjoong Jang, Andreas Meuleman, Dahyun Kang, Donggun Kim, Christian Richardt, and Min H Kim. Egocentric scene reconstruction from an omnidirectional video. *ACM Transactions on Graphics (TOG)*, 41(4):1–12, 2022. 1, 2

[17] Yoonwoo Jeong, Seokjun Ahn, Christopher Choy, Anima Anandkumar, Minsu Cho, and Jaesik Park. Self-calibrating neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5846–5854, October 2021. 8

[18] Akira Kageyama and Tetsuya Sato. "yin-yang grid": An overset grid in spherical geometry. *Geochemistry, Geophysics, Geosystems*, 5(9), 2004. 2, 3

[19] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5741–5751, October 2021. 8

[20] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *Advances in Neural Information Processing Systems*, 33:15651–15663, 2020. 3

[21] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. 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 (CVPR)*, pages 7210–7219, June 2021. 3

[22] 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. *Communications of the ACM*, 65(1):99–106, 2021. 2, 3, 4, 5, 6, 7

[23] Pierre Moulon, Pascal Monasse, Romuald Perrot, and Renaud Marlet. OpenMVG: Open multiple view geometry. In *International Workshop on Reproducible Research in Pattern Recognition*, pages 60–74. Springer, 2016. 6

[24] 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. 3, 5

[25] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11453–11464, June 2021. 3

[26] Ryan S Overbeck, Daniel Erickson, Daniel Evangelakos, Matt Pharr, and Paul Debevec. A system for acquiring, processing, and rendering panoramic light field stills for virtual reality. *ACM Transactions on Graphics (TOG)*, 37(6):1–15, 2018. 1, 2

[27] Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5865–5874, October 2021. 3

[28] Albert Parra Pozo, Michael Toksvig, Terry Filiba Schrager, Joyce Hsu, Uday Mathur, Alexander Sorkine-Hornung, Rick Szeliski, and Brian Cabral. An integrated 6dof video camera and system design. *ACM Transactions on Graphics (TOG)*, 38(6):1–16, 2019. 1, 2

[29] Leonid I Rudin and Stanley Osher. Total variation based image restoration with free local constraints. In *Proceedings of 1st international conference on image processing*, volume 1, pages 31–35. IEEE, 1994. 5

[30] Pratul P. Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T. Barron. Nerv: Neural reflectance and visibility fields for relighting and view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7495–7504, June 2021. 3

[31] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5459–5469, June 2022. 2, 3, 5, 6, 7

[32] Yule Sun, Ang Lu, and Lu Yu. Weighted-to-spherically-uniform quality evaluation for omnidirectional video. *IEEE signal processing letters*, 24(9):1408–1412, 2017. 6

[33] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 6

[34] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. Nerf–: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021. 8

[35] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5752–5761, 2021. 3

[36] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 6