VR-NeRF: High-Fidelity Virtualized Walkable Spaces

Linning Xu
The Chinese University of Hong Kong
Hong Kong
Meta
USA

Vasu Agrawal
Meta
USA

William Laney
Meta
USA

Tony Garcia
Meta
USA

Aayush Bansal
Meta
USA

Changil Kim
Meta
USA

Samuel Rota Bulò
Meta
Switzerland

Lorenzo Porzi
Meta
Switzerland

Peter Kontschieder
Meta
Switzerland

Aljaž Božič
Meta
Switzerland

Dahua Lin
The Chinese University of Hong Kong
Hong Kong

Michael Zollhöfer
Meta
USA

Christian Richardt
Meta
USA

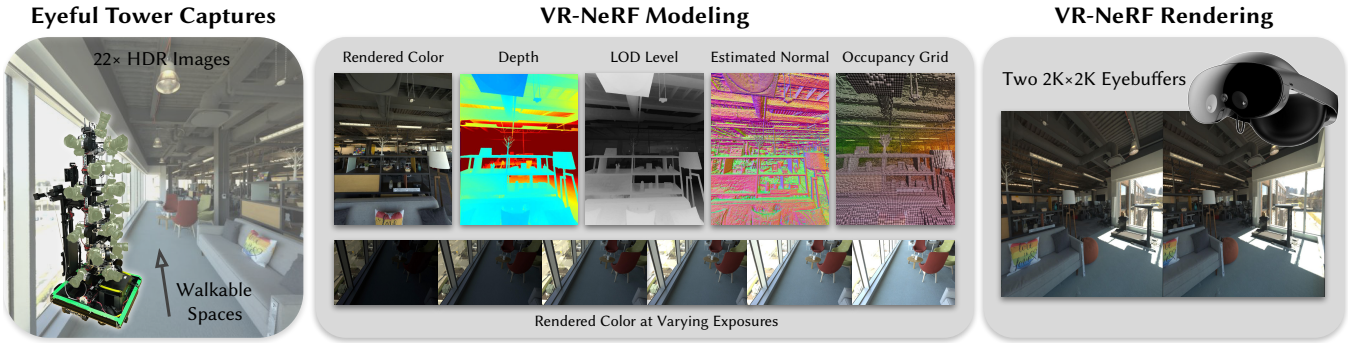


Figure 1: VR-NeRF brings high-fidelity walkable spaces to real-time virtual reality. Our "Eyeful Tower" camera rig captures spaces with high image resolution and dynamic range that approach the limits of the human visual system. We train high-fidelity neural radiance fields that exploit the high-dynamic range nature of our captured scenes and provide level-of-detail mip-mapping for efficient anti-aliasing. Our rendering backend leverages our accurate occupancy grid and a dynamic multi-GPU work distribution scheme to achieve real-time frame rates on dual 2K×2K eyebuffers for an immersive VR experience.

ABSTRACT

We present an end-to-end system for the high-fidelity capture, model reconstruction, and real-time rendering of walkable spaces in virtual reality using neural radiance fields. To this end, we designed and built a custom multi-camera rig to densely capture walkable spaces in high fidelity and with multi-view high dynamic range images in unprecedented quality and density. We extend instant neural graphics primitives with a novel perceptual color space for learning accurate HDR appearance, and an efficient mip-mapping mechanism for level-of-detail rendering with anti-aliasing, while carefully optimizing the trade-off between quality and speed. Our multi-GPU renderer enables high-fidelity volume rendering of our neural radiance field model at the full VR resolution of dual 2K×2K at 36 Hz on our custom demo machine. We demonstrate the quality of our results on our challenging high-fidelity datasets, and compare our method and datasets to existing baselines. We release our dataset on our project website: https://vr-nerf.github.io.

SA Conference Papers '23, December 12–15, 2023, Sydney, NSW, Australia

ACM ISBN 979-8-4007-0315-7/23/12.
https://doi.org/10.1145/3610548.3618139

KEYWORDS

Multi-View Capture, Neural Radiance Fields, Novel-View Synthesis, High Dynamic Range Imaging, Real-Time

ACM Reference Format:
Linning Xu, Vasu Agrawal, William Laney, Tony Garcia, Aayush Bansal, Changil Kim, Samuel Rota Bulò, Lorenzo Porzi, Peter Kontschieder, Aljaž Božič, Dahua Lin, Michael Zollhöfer, and Christian Richardt. 2023. VR-NeRF: High-Fidelity Virtualized Walkable Spaces. In *SIGGRAPH Asia 2023 Conference Papers (SA Conference Papers '23), December 12–15, 2023, Sydney, NSW, Australia.* ACM, New York, NY, USA, 23 pages. https://doi.org/10.1145/3610548.3618139

1 INTRODUCTION

The advent of consumer virtual reality (VR) headsets has led to a proliferation of highly immersive visual media, including breath-taking VR photography and video. However, existing approaches support either high-fidelity view synthesis with a small *headbox* of less than 1 m diameter [Broxton et al. 2020; Overbeck et al. 2018], or scene-scale free-viewpoint view synthesis of lower quality or frame-rate [Jang et al. 2022; Parra Pozo et al. 2019; Wu et al. 2022a]. In this work, we present a comprehensive system designed to overcome these limitations all the way from capture to rendering for high-fidelity free-viewpoint exploration of walkable, real-world static spaces in VR. Our contributions address the following challenges:

(1) dense, high-fidelity capture of large-scale walkable spaces,
(2) high-fidelity neural radiance field reconstruction, and
(3) real-time rendering of our neural radiance fields in VR.

High-fidelity view synthesis depends on high-quality, densely captured multi-view images. While NeRF objects use 100s of views [Mildenhall et al. 2020] and light field captures around 1,000 views per location [Broxton et al. 2020], walkable scenes will need a minimum of several thousand input views to provide enough spatial coverage. Existing captures of walkable spaces tend to be hand-held and usually comprise 100s of photos [e.g. Philip et al. 2021] or 1,000s of video frames [e.g. Knapitsch et al. 2017]. In both cases, the space of camera poses is undersampled: photo sequences lack sufficient density, and videos move along a 1D subspace that fails to sample the 6D pose space sufficiently uniformly. High-fidelity view synthesis also needs to reproduce the high dynamic range of the real world, which existing methods do not. To this end, we designed a custom camera rig that enables capturing walkable spaces in unprecedented quality and density: our datasets contain thousands of 50 megapixel high dynamic range (HDR) images. Several of our datasets exceed 100 gigapixels – two orders of magnitude more than existing datasets [Flynn et al. 2019; Philip et al. 2021; Xu et al. 2021].

Neural radiance fields (NeRFs) have led to an explosion in high-quality novel-view synthesis techniques [Mildenhall et al. 2020; Tewari et al. 2022]. However, existing methods do not support the size, scale, and dynamic range of our high-fidelity datasets, even when downsampled to 2K resolution. We propose VR-NeRF, which is uniquely adapted to our high-quality datasets and supports real-time VR rendering in full NeRF quality. Specifically, we introduce a new perceptually based color space for representing high-dynamic range radiance values of up to 10,000 cd/m^2, allowing our model to learn up to 22 stops[1] of dynamic range (or 4,194,304:1). A second crucial component is a real-time-capable mip-mapping technique that suppresses aliasing when observing objects at different distances using level-of-detail rendering. We also developed a principled pruning stage to obtain an accurate occupancy grid for speeding up rendering with a focus on improved geometry estimation.

The third and final stage of our end-to-end system is a custom multi-GPU renderer that brings high-fidelity NeRF rendering into virtual reality. On our custom-built demo machine, we can render our models at the full resolution of the Quest Pro VR headset, i.e., two 2K×2K eye buffers (~8 megapixel), at a consistent frame rate of 36 Hz, which results in a compelling VR experience that enables free exploration of walkable spaces in high fidelity.

2 RELATED WORK

Kanade et al. [1995] coined the term "Virtualized Reality" to see a previously recorded event from any perspective. Our goal is to virtually walk through previously captured scenes at high fidelity in virtual reality. We, therefore, call our work *Virtualized Walkable Spaces*. There are three crucial components to enable high-fidelity virtualized walkable spaces: (1) a mobile high-resolution multi-view camera system to densely capture large-scale scenes; (2) an efficient neural representation to compactly and accurately encode a large-scale scene with high dynamic range and level of detail; and (3) optimized real-time rendering at VR resolution and frame rate.

High-Resolution Multi-View Capture System. Capture systems can vary from a single moving camera [Bertel et al. 2020; Davis et al. 2012; Gortler et al. 1996; Hedman et al. 2016; Kim et al. 2013; Knapitsch et al. 2017; Levoy and Hanrahan 1996] to multi-camera rigs [Broxton et al. 2020; Flynn et al. 2019; Parra Pozo et al. 2019; Wilburn et al. 2005] and synchronized camera arrays in big studios [Joo et al. 2019; Orts-Escolano et al. 2016]. Existing multi-view captures are either limited to a small headbox [e.g. Overbeck et al. 2018; Parra Pozo et al. 2019] or are sparsely captured [e.g. Knapitsch et al. 2017; Yoon et al. 2020], which restricts freedom of motion. We built a multi-camera rig that densely and efficiently captures a wide variety of walkable spaces to create large-scale multi-view datasets with high-resolution details (50 megapixels) and high dynamic range.

Large-scale Novel View Synthesis. Our focus is on real-time VR rendering of high-fidelity walkable spaces; recent surveys cover the full range of scene representations [Richardt et al. 2020; Tewari et al. 2022]. While mesh-based reconstructions [Straub et al. 2019; Whelan et al. 2018] are ideal for fast rendering, they tend to lack fine geometric detail. Image-based rendering [e.g. Hedman et al. 2016] achieves more visual detail but struggles with reflective surfaces. Several follow-up methods use neural representations for explicit reflection support [Philip et al. 2021; Wu et al. 2022a; Xu et al. 2021] and achieve interactive frame rates. NeRFs [Mildenhall et al. 2020] have become the de-facto standard neural representation due to their versatility and ability to represent complex scenes with high fidelity. They have been extended in multiple ways to represent large-scale scenes even at a city scale [Tancik et al. 2022; Turki et al. 2022; Xiangli et al. 2022; Xu et al. 2023; Zhang et al. 2023]. High-resolution concerns have also been addressed [Jiang

[1] One stop is a doubling or halving of the amount of light reaching the imaging sensor.

et al. 2023; Wang et al. 2022]. However, these methods do not support level of detail and high dynamic range, which are required for high-fidelity VR. LocalRF [Meuleman et al. 2023] and F^2-NeRF [Wang et al. 2023] tackle large unbounded scenes, yet only support limited view extrapolation and thus cannot provide fully immersive free-view exploration. Methods built on implicit surfaces, like signed distance functions, tend to focus on high-quality 3D surface reconstruction rather than view synthesis [Li et al. 2023; Rosu and Behnke 2023; Yu et al. 2022; Zhu et al. 2023]. We build our model on Instant-NGP (iNGP) [Müller et al. 2022], as it supports real-time rendering without model baking, and provides easily extensible model capacity via its hash grid. However, it lacks support for high-fidelity rendering of large-scale walkable spaces, such as level of detail and perceptually based HDR support.

High Dynamic Range (HDR). The human visual system supports a significantly higher dynamic range than current camera or display technology [Reinhard et al. 2006]. When recreating highly realistic walkable spaces, it is therefore important to accurately capture and render the scene in HDR. RawNeRF learns linear radiance from raw sensor measurements using a weighted L2 loss that approximates a tonemapped loss [Mildenhall et al. 2022]. Several methods learn to reconstruct linear radiance from low dynamic range images using differentiable tonemapping models [Huang et al. 2022; Jun-Seong et al. 2022; Rückert et al. 2022]. We train our HDR model directly using HDR input images in a novel perceptually uniform color space that does not require custom losses or tonemapping modules.

Level of Detail (LOD). Takikawa et al. [2021] and Barron et al.'s Mip-NeRF [2021] introduced the notion of level of detail into neural signed distance and radiance fields, respectively, to reduce geometric and visual complexity, e.g. to minimize aliasing when viewing objects from a distance. As Mip-NeRF's integrated positional encoding is incompatible with efficient grid-based NeRF approaches like iNGP [Müller et al. 2022], Zip-NeRF [Barron et al. 2023] uses supersampling as an approximation, but multi-second inference times still prevent real-time rendering. Aroudj et al. [2022] store the scene redundantly at multiple LOD levels in a sparse voxel octree. We introduce an efficient LOD approach designed for iNGP that enables high-fidelity real-time VR rendering with anti-aliasing.

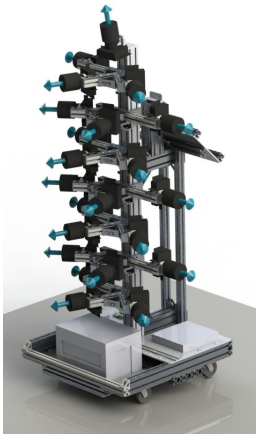
3 THE "EYEFUL TOWER" CAPTURE RIG

Capturing scenes with a hand-held camera quickly reaches limits: taking hundreds of photos is tedious, achieving consistent coverage of viewpoints is difficult, and hand-held exposure bracketing is tricky due to camera shake. To capture real-world environments with the highest visual fidelity in terms of spatial resolution and dynamic range, we designed, built, and refined a custom multi-camera capture rig affectionately referred to as the *Eyeful Tower*. The design of our capture rig was guided by the following considerations:

(1) *Coverage:* Place cameras for approximately uniform light field capture, and parallelize data capture across cameras.
(2) *Fidelity:* Match human visual perception in terms of acuity and high dynamic range.
(3) *Mobility:* Allow single-person operation, and be usable without external power or network connection.
(4) *Rigidity:* Support multi-exposure bracketing for high dynamic range (HDR) reconstruction without camera motion.
(5) *Storage:* Record photos on-camera, so no server is needed. Offload all photos via a single network cable.

3.1 Capture Rig Design

We built our capture rig using extruded aluminium around an 80×80 cm base with a 1.8 m vertical pole for 22 cameras that are distributed on 7 levels with 3 cameras each, plus one upward-facing camera at the top (see right). A 1.5 kWh Li-ion battery powers cameras, a 24-port network switch, and a Raspberry Pi controlling the cameras. We chose Sony α1 mirrorless cameras for their high-quality 50-megapixel raw images with 14 stops of dynamic range. Please see our supplement for details on the rig design and camera/lens choices.



3.2 Capture Process

3.2.1 Desirable Capture Density. Reproducing the appearance of a static scene from any viewpoint in theory requires observations for the entire 5D plenoptic function [Adelson and Bergen 1991]. The widely used NeRF synthetic dataset [Mildenhall et al. 2020] has viewpoints densely distributed on a hemisphere, which allows the renderings to generalize continuously across the whole hemisphere of viewing directions. For scene-scale rendering, we are lacking such a densely captured dataset, which results in the *limited capability to extrapolate novel viewpoints*. However, this is critically important for virtual reality, where we want to deliver walkable spaces with 6-degrees-of-freedom allowable head movement.

3.2.2 Capture Procedure. We capture scenes by 'tiling' the available floor area with rig positions that are spaced roughly 30 cm apart. For complete captures, we capture forward- and backward-facing views, while trying to stay at least 30 cm away from walls or objects. Near walls, it is often sufficient to only capture the direction facing away from the wall, as defocused close-ups of a wall usually add little value. Before each capture, we also place scale bars (for automatic scale estimation) and a Macbeth ColorChecker (for color verification and white balance) into the scene. During the capture, we try to stay out of view of any camera, avoid moving any objects, such as chairs or carpets, and aim to minimize lighting changes and shadow casting.

3.3 Data Preprocessing

3.3.1 HDR Image Merging. We use LibRaw 0.21 to debayer the raw images captured by our cameras to 16-bit linear TIFF images. We then merge 9 different exposures into one high-dynamic range image using a robustified version of Hanji et al.'s Poisson photon noise estimator [2020], which provides an unbiased estimate of scene radiance. We observed that the Sony α1 raw image values do not saturate as quickly as expected, which produces outliers that can reduce the estimated radiance sufficiently to cause visible

color changes. Therefore, we keep track of the minimum radiance estimate per pixel and color channel, so that we can ignore it when merging the input exposures. In addition, we set fully saturated pixels to the lowest radiance that saturates in all images.

3.3.2 Camera Calibration. We estimate camera poses and intrinsics using Agisoft Metashape Pro 2.0 [Agisoft, LLC 2023], a professional photogrammetry software that supports rig calibration, in which the relative pose between cameras is constant across all positions of the capture rig within a scene. Metashape effectively handles our large-scale datasets with up to 6,300 photos at 50 megapixel resolution [Over et al. 2021]. It also automatically detects the markers on our calibrated scale bars, such that camera poses are in metric space for 1:1 scale rendering in VR.

3.3.3 Captured Datasets. We captured multiple datasets using our Eyeful Tower capture rig, which are summarized in Table 1. Our captures took between 5 minutes and 6 hours, depending on the scale and complexity of the scene, with an average speed of around one minute per m^2. The resulting datasets comprise 29–303 billion pixels, or rays, covering spaces of 6–120 m^2.

Table 1: Statistics of scenes captured using our Eyeful Tower rig: We show the number of cameras, rig positions, and images, as well as the capture time, surface area, and the number of rays at full resolution (5,784×8,660) and 1368×2048 ('2K'), our typical training and rendering image resolution.

Scene	Cameras	# Pos.	# Img.	Time	Area	Rays	Rays @ 2K
APARTMENT	22	180	3,960	60 min	55 m^2	190.6 B	10.7 B
KITCHEN	19	318	6,024	43 min	54 m^2	302.7 B	16.9 B
OFFICE1A	9	85	765	23 min	20 m^2	29.1 B	1.6 B
OFFICE1B	22	71	1,562	16 min	20 m^2	78.2 B	4.4 B
OFFICE2	9	233	2,097	39 min	35 m^2	79.8 B	4.5 B
OFFICE_VIEW1	22	126	2,772	31 min	18 m^2	138.9 B	7.8 B
OFFICE_VIEW2	22	67	1,474	10 min	33 m^2	73.8 B	4.1 B
RIVERVIEW	22	48	1,008	5 min	6 m^2	52.9 B	3.0 B
SEATING_AREA	9	168	1,512	22 min	16 m^2	55.9 B	3.1 B
TABLE	9	134	1,206	14 min	24 m^2	45.2 B	2.5 B
WORKSHOP	9	700	6,300	364 min†	120 m^2	239.4 B	13.4 B

† Includes 121 minutes of capture time and 243 minutes of data offload mid-capture.

4 HIGH-FIDELITY NEURAL RADIANCE FIELDS

Volume rendering using neural radiance fields is a compelling choice for photorealistic scene representations due to the versatility of representing semi-transparent surfaces and finely detailed objects while being suitable for delivering scene-scale rendering. As our focus is on maximizing rendering fidelity in the available compute budget, we use neural radiance fields as a foundation and leave alternative representations as future work. Instead of constructing a large, complex model with extra capacity to account for various effects, our goal is to design a simple yet general model that facilitates real-time VR rendering for large-scale scenes.

We therefore build on the Instant NGP architecture [Müller et al. 2022] with its efficient and *scalable* multi-level hash encoding for *fast* rendering of large-scale static scenes. We make several contributions to improve the visual fidelity of high-resolution room-scale rendering, including a perceptual color space that enables perceptual optimization of high dynamic range images using a simple L_1 loss. We further introduce an efficient and effective level-of-detail scheme for anti-aliasing using multi-level hash grids. To faithfully represent unbounded areas, such as views through windows or long corridors, we adopt a cubic space contraction based on the L_∞ norm [Wan et al. 2023], which is a good fit for grid-based representations. We discuss implementation details and additional components that contribute to the high quality of our view synthesis model in our supplement.

4.1 Perceptual Modeling of High Dynamic Range

Our Sony α1 cameras capture raw images with a dynamic range of 14 stops (i.e., 14 bits of usable information). The 9-step exposure bracketing adds a further 8 stops, for a total of 22 stops of dynamic range (see Figure 2). In other words, the brightest input pixel value can be up to 4,194,304 times as bright as the darkest non-zero pixel value. Applying common image losses like L_1 or L_2 directly in linear color spaces of this range leads to poor results as the losses are dominated by errors in bright areas. For example, an error of 0.1 is significantly more noticeable at a base level of 0.1 (+100%) compared to 10 (only +1%), yet would be penalized the same. The solution is to either use a more complex loss function, such as RawNeRF's relative MSE [Mildenhall et al. 2022], or a carefully designed non-linear mapping to a perceptually uniform color space.

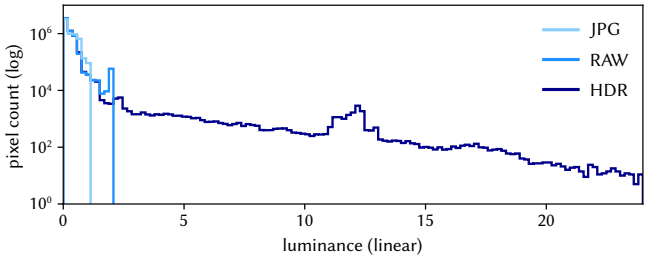


Figure 2: Comparison of the dynamic range of a JPEG photo (range 0~1) with the corresponding raw image (0~2) and the full HDR image (0~145).

One such non-linear mapping is the Perceptual Quantizer (PQ) developed by Dolby [Miller et al. 2013] and standardized by SMPTE [2014], which is the foundation of many consumer HDR image and video formats. PQ was designed to optimally encode the large luminance range from 0 to 10,000 cd/m^2 in 10–16 bits while minimizing visible banding artifacts. This was achieved by approximating the integral of just noticeable differences based on the contrast sensitivity function of the human visual system [Kunkel 2022]. The function

$$\mathrm{PQ}(Y) = \left(\frac{c_1 + c_2 \cdot Y^{m_1}}{1 + c_3 \cdot Y^{m_1}}\right)^{m_2} \quad \text{with constants} \tag{1}$$

$$m_1 = \frac{1305}{8192},\ m_2 = \frac{2523}{32},\ c_1 = \frac{107}{128},\ c_2 = \frac{2413}{128},\ c_3 = \frac{2392}{128} \tag{2}$$

maps the input luminance $Y \in [0, 10{,}000]$ cd/m^2 to the 'PQ space' in the unit range. For our experiments, we map linear color values of 1 to a luminance of 100 cd/m^2 in order to allow a conversion to

the PQ space. Operating in the unit range is also a natural fit for the sigmoid activation function, which eases the learning of model output distributions. Applying an L_1 or L_2 loss in the PQ space now penalizes errors according to human visual perception, and is able to produce colors in full high dynamic range. See Figure 7 for a sweep of different exposures at rendering time.

4.2 Feature Grid Mip-Mapping for Level-of-Detail

Level-of-detail (LOD) rendering is desirable for large-scale scenes, as objects observed at different distances reveal varying levels of geometric and texture detail. Single-LOD methods like NeRF or iNGP can cause severe aliasing in highly textured objects seen at a distance, while details seen in only a few views might be washed out due to many overlapping distant views. Multiple levels of detail can reduce aliasing as the LOD level can be dynamically adjusted based on the distance of objects from the viewer. In computer graphics, texture LOD is usually implemented using mip-maps [Williams 1983]. Mip-NeRF [Barron et al. 2021] introduced mip-mapping to NeRFs and Zip-NeRF [Barron et al. 2023] recently extended these ideas to fast grid-based feature encodings, as used by iNGP. Unfortunately, this approach is unsuitable for real-time rendering (1.1 FPS on 8×V100). Instead, we introduce a simple but effective mip-mapping scheme for grid-based feature encodings that enables learning of continuous LOD while actively supporting real-time rendering.

4.2.1 Feature Grid Mip-Mapping. Multi-resolution feature grids are a natural fit for LOD rendering as they already represent features across multiple scales. By considering a ray as a cone as in Mip-NeRF, and by comparing its cross section with the size of grid features at each level, we can efficiently determine which feature grid levels are theoretically resolvable at the ray level, and can down-weight or even ignore finer levels that would introduce aliasing.

For a specific ray, we start by calculating its base radius r at unit distance along the ray. At a sample location, the pixel footprint is then determined by multiplying the base radius with the metric distance t along the ray as $\hat{r} = t \cdot r$. For contracted spaces, Barron et al. [2022, 2023] and Wang et al. [2023] consider the Jacobian $\mathbf{J}_C$ of the contraction function $C(\cdot)$ at the sample location $\mathbf{x}$ to calculate the scale factor for variance or step size estimation. Similarly, we could derive the contracted pixel radius via $C(\hat{r}) = \hat{r} \cdot \sqrt[3]{\det(\mathbf{J}_C(\mathbf{x}))}$. In practice, we compute the contracted pixel radius directly from corresponding sample points on adjacent rays in the contracted space. The optimal LOD level can then be calculated from the configuration of the multi-resolution feature grid as follows. Suppose the base resolution is s and the scale factor between levels is f. For the L^{th} level (with $L = 0$ being the base), each level has a grid voxel size of $(sf^L)^{-1}$. Based on the Nyquist–Shannon sampling theorem, we dampen features whose size is less than twice the footprint $\hat{r}$ in the contracted coordinate space (see diagram in Figure 3). The optimal LOD level for a sample is therefore $L^* = -\log_f(2s\hat{r})$. For a

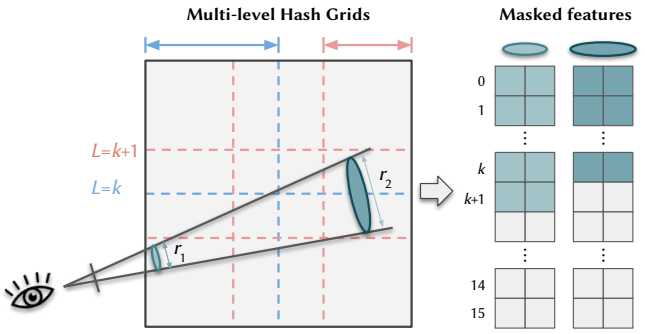


Figure 3: LOD Masking. Smaller sample footprints like r_1 return more features from the multi-level hash grid than larger, more distant samples (r_2).

piecewise linear LOD transition, we use these per-level weights:

$$w_L = \begin{cases} 1 & L \le \lfloor L^* \rfloor \\ L^* - \lfloor L^* \rfloor & \lfloor L^* \rfloor < L \le \lceil L^* \rceil \\ 0 & \lceil L^* \rceil < L \end{cases} \tag{3}$$

For distant points, we only need to sample the features of the lowest few grid levels, which reduces rendering time, while gradually revealing high-frequency features for closer points. Feature sampling of the finest hash grid layers is particularly expensive due to the highly incoherent memory access patterns. Skipping these features results in substantially faster rendering (Section 5).

4.2.2 LOD Bias. Similar to standard mip-mapping, we can optionally add an LOD bias ΔL to the LOD L^* used for querying and weighting the grid features. This continuously adjusts the sharpness of details to balance between blurred and aliased rendering. In fact, the LOD bias can be viewed as a unifying framework that encompasses coarse-to-fine training strategies [Lin et al. 2021; Park et al. 2021; Yang et al. 2023]. Such progressive training approaches start with low-frequency models and gradually increase the number of feature scales to improve details. This is equivalent to starting with a large negative LOD bias, such that only low-frequency features are used, and annealing it towards zero during training. See Figure 8 for the visualized LOD bias sweep on the APARTMENT scene.

4.2.3 Distance-aware Features. By mip-mapping grid features, we are effectively making the features used for radiance computation distance-aware, as different features are used at different viewing distances. This offers an additional degree of freedom to handle inconsistent data during the capture process, such as the distance-dependent shadows cast by the camera rig. Rig shadows are most prominent when the rig approaches walls or corners. With limited training views in these ambiguously captured locations, the model is likely to fake the shadows with incorrect geometry and/or appearance. On the other hand, distance-aware features allow our model to learn distance-dependent appearance, which reduces visual artifacts. We further noticed that the mip-mapped features encourage the model to better allocate model capacity for fine-grained details.

4.3 Optimizing the Quality–Speed Trade-off

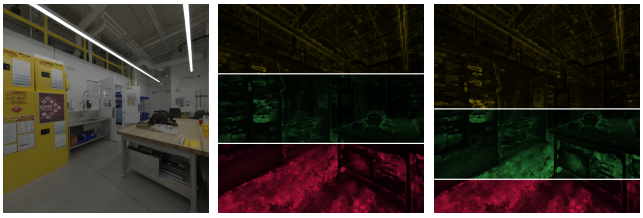
Our goal of high-fidelity real-time NeRF rendering requires some challenging trade-offs between visual quality and rendering speed. For example, while conditional latent codes and wider and deeper networks can improve rendering quality [Barron et al. 2023; Müller et al. 2022], they come at a significant run-time cost. Similarly, using a proposal network for sampling adds overhead at render time as multiple networks need to be evaluated sequentially. To maximize rendering speed without model baking, we implement an explicit binary occupancy grid for efficiently skipping free space and minimizing the number of sample points for which hash grid features need to be queried and MLPs evaluated. An example grid is shown in Figure 9, along with the corresponding image results. While occupancy grids are widely-adopted acceleration structures [Chen et al. 2022; Liu et al. 2020; Müller et al. 2022; Sun et al. 2022], we propose two novel extensions that help us prune more accurately.

4.3.1 Cylinder Pruning. We initialize our binary occupancy grid based on the known rig capture positions to carve out as much free space in the scene as possible. For this, we first approximate the geometry of our Eyeful Tower capture rig as a cylinder. We then mark all occupancy grid voxels that are completely inside any such cylinder as free space. This type of pruning has two key benefits: (1) it prevents the model from cheating using floaters in front of cameras, which leads to more view-consistent models, and (2) it speeds up the early stages of training and thus helps improve convergence speed.

4.3.2 Joint History- and Grid-based Pruning. We explore a more conservative pruning strategy that combines pruning based on training history with dense grid sampling. History pruning keeps track of the maximum density observed for each voxel in the occupancy grid during the training process. This only considers rays seen during training, so some parts of the scene may not be observed. Grid-based pruning makes up for this by evaluating a dense cubic grid inside each voxel of the occupancy grid to estimate the maximum density for each voxel. As the density depends on the step size used in training, we use a worst-case estimate for this, i.e., the minimum step size possible inside each voxel based on the ray from the closest camera. For our datasets, we start the pruning process after 100K iterations, when a relatively clean scene geometry is obtained. Every thousand iterations, we prune grid voxels for which both maximum densities fall below the current pruning threshold (which we anneal linearly from zero to $\alpha = 0.2$). We start with a coarse occupancy grid of 128^3 resolution, and upsample the occupancy grid at predefined iteration milestones to prune scenes more accurately over time.

5 VR NERF RENDERING

Rendering a room-scale NeRF model in VR requires high resolution, high frame rates and low latency. Our target is native rendering on a Meta Quest Pro VR headset, ideally dual 2K×2K eyebuffers at 72 FPS. We approach this task with a combination of hardware, software, and model optimizations. Specifically, we present a custom multi-GPU CUDA renderer with efficient in-register MLP evaluation and automatic work distribution, a compute-efficient LOD technique (see Section 4.2), and a 20-GPU workstation for peak VR performance.



(a) Color image (b) Static split: 49.3 FPS (c) Dynamic split: 73.6 FPS

Figure 4: (a) In this example, we render a novel view using 3 GPUs. (b) A static split distributes work equally (indicated by colors; brightness is proportional to #MLP evaluations). (c) Our dynamic work split achieves 49% higher FPS.

MLP evaluation is the most computationally expensive portion of model inference, and thus a prime candidate for optimization. Our MLP implementation is specialized for small iNGP-style networks by taking advantage of Nvidia's Tensor Cores and evaluating all layers within registers. Inputs and outputs of the MLP are stored in shared memory while per-layer activations are stored in register-backed arrays, with outputs from one layer being shuffled in an architecture-dependent way to become the inputs to the next layer. This limits memory traffic to just the input and output features, which are typically small (32 inputs, 16 bottleneck features, 3 output colors) compared to the hidden layers (64 nodes), and the network weights, which are shared across the kernel and typically cached. This structure also allows the MLP evaluation to be interleaved with ray marching and hash grid sampling in a single kernel. This enables the neural features to be passed to the networks without staging through global memory (which can suffer from capacity problems with a large number of samples per ray) or across multiple kernels (which would incur extra launch and synchronization overhead).

We further adopt a dynamic work distribution strategy for improving the utilization of multiple GPUs compared to a static work split that would often be suboptimal as some rays take longer to compute than others due to differences in pruning in different parts of the scene, as well as GPU caching and overhead effects. For every frame, we measure the throughput per GPU in rays per second, and assign contiguous rows to each GPU based on its ratio of the total throughput. We use dampening for smoother convergence to an optimal distribution. Figure 4 demonstrates a 49% increase in FPS. Each GPU stores a separate copy of each scene (~700 MB VRAM).

We also built a custom 20-GPU rendering workstation to evaluate our walkable spaces at the highest possible fidelity in virtual reality. This machine comprises a Dell R7515 server with an AMD Epyc 7313P CPU and 256 GB of RAM, and is connected to 20 Nvidia A40 GPUs via a PCIe switching solution from Liqid Inc., all in a 24U server rack. We detail our design considerations in the supplement.

6 RESULTS AND EVALUATION

For our room-scale scenes, we use hash grid configurations with $L = 16$ levels of two features, with a base resolution of 128 and scaling

Table 2: Quantitative comparison results on the Eyeful Tower test set. All results are trained on 1K resolution images for 110K iterations with 1024 samples per ray. We report the average PSNR/SSIM/LPIPS both in sRGB and PQ color spaces. The best results are highlighted. See the supplemental document for the breakdown by individual dataset.

Methods	PSNR ↑	SSIM ↑	LPIPS ↓	PQ-PSNR↑	PQ-SSIM↑	PQ-LPIPS↓
iNGP (our implementation)	31.93	0.918	0.183	37.39	0.957	0.133
with PQ color space	32.47	0.926	0.170	38.15	0.962	0.122
with PQ color space and LOD	33.30	0.930	0.146	38.95	0.964	0.108

factor of 1.4. Following iNGP, we use a 1-hidden-layer density MLP and a 2-hidden-layer color MLP, both 64 neurons wide. For each ray, we sample 1024 points using exponential distances for integration. We use the Adam optimizer [Kingma and Ba 2015] with $\beta_1 = 0.9$, $\beta_2 = 0.99$, $\epsilon = 10^{-15}$, and a batch size 12,800 rays (256 random rays from 50 random images) for all our experiments. We use far-field contraction for the subset of unbounded scenes. We use learning rate 0.01 for the hash grids and 0.005 for the remaining modules. We discuss a series of additional techniques for per-scene quality improvements in the supplement.

For fair evaluation, we hold out a fixed camera from the training set, which has the same number of frames as all other cameras. For ablation experiments, we show results trained for 110K iterations on 1K resolution Eyeful Tower datasets. The demo videos are produced by models trained on 2K resolution images and longer than 200K iterations. In the supplement, we include additional results on the Inria [Philip et al. 2021] and mip-NeRF360 datasets [Barron et al. 2022], as well as ablations on pruning strategies.

6.1 Comparative Evaluation

To model HDR images, iNGP [Müller et al. 2022] suggests using an exponential color activation for linear RGB space. RawNeRF [Mildenhall et al. 2022] further suggests using a weighted loss to prevent extremely bright areas from dominating. Table 2 and Figure 6 show the comparisons of our designed modules with iNGP baselines: (1) the effectiveness of using PQ color space for HDR modeling, and (2) the use of the LOD feature grid.

We choose the baseline of using iNGP with linear color space with truncated exponential activation for the color network to avoid the issue of exploding values weighted by the predicted color value, as practiced by Mildenhall et al. [2022]. "iNGP with PQ color space" reflects our modification of directly training in the PQ color space, and replaces the original exponential color activation with a sigmoid function. "iNGP with PQ color space and LOD" represents our core model of adopting mip-mapped grid features based on the estimated LOD level for each queried sample point on the ray. We report the standard PSNR/SSIM/LPIPS metrics in tonemapped sRGB space, and additionally report versions of these metrics in PQ color space for better evaluation on extremely bright and dark areas. More results and analysis with supporting plots and visualizations on each ablated module can be found in our supplement.

PQ color space. Table 2 shows that the PQ color space consistently outperforms the linear color space for all test scenes and all metrics. We noticed that during training, color predictions using the exponential activation baseline continue to grow to excessively large values. This poses an ambiguity for predicting correct density values and their derived weights, which are multiplied with the point color to obtain sample colors. Directly modeling colors in linear RGB space poses additional challenges in regressing and interpolating accurate color values, especially when a large range of radiance is present. As shown in the second example in Figure 6, the base model fails to model the color on the checkerboard correctly.

Mip-mapped features. The combination of LOD and PQ color space further improves the rendering quality and leads to cleaner geometries, as seen in Table 2 and Figure 6. This is critical for pruning and VR rendering, where a good geometry is desired. Figure 6 shows four scenarios where the mip-mapped features can help. The first scene shows an annoying dark appearance baked into the rendered scene due to dynamic shadows from the rig in the training data. These are modeled by the high-frequency levels and well addressed by the distance-aware features. The second example shows both cleaner geometry and appearance in the ambiguous space near the whiteboard. The third example shows how LOD helps eliminate aliasing for distant areas, especially when observing the scene from a wide angle. The last example shows how LOD can also reveal more detail compared to non-mip-mapped features, not just a cleaner appearance. This is expected as the features corresponding to high-resolution hash grids are specifically allocated to areas rich in fine details in the training views, which allows the model to automatically allocate more capacity for these parts. Note that while the quantitative metrics are similar to the baselines, the visual improvements are easier to spot and critical to the high-fidelity rendering results that contribute to a pleasant VR experience.

6.2 Performance Evaluation

Figure 5 plots the rendering frame rate when using a varying number of A40 GPUs. Native VR rendering for a Meta Quest Pro headset requires rendering two 2064×2096 eyebuffers at 72 FPS. However, Asynchronous Spacewarp (ASW) [Beeler et al. 2016] can help close this gap by reprojecting frames when they are rendered at least at half the native FPS, i.e., 36 (dashed line) instead of 72 (solid line). But even ASW fails if rendering is slower than that critical threshold. Thus, we found the 'p99' metric (the 99$^{\text{th}}$ percentile of FPS) to be a better proxy for the quality of VR experience than mean FPS — as long as the application is typically (i.e. 99%+ of the time) above 36 FPS, ASW can deliver a smooth experience. Any slower, and the user may notice stuttering frames, and experience motion sickness.

An off-the-shelf 3-GPU workstation is sufficient for reliable half-resolution VR rendering of half the scenes at 36 FPS (see Figure 5, top right), which demonstrates the practicality of our method. For maximum rendering speed and fidelity, we use our custom 20-GPU rendering workstation. All scenes but one achieve a p99 of 72 FPS at half-resolution, and thus provide a smooth VR experience even without ASW. At full resolution (top row in Figure 5), 4 of the 6 scenes shown in Figure 5 (bottom right) exceed the critical ASW threshold of 36 FPS for a smooth, high-fidelity VR experience. Interestingly, halving the resolution only approximately doubles the FPS, even though only 1/4 as many pixels are being rendered. This may indicate a substantial amount of per-frame overhead (e.g.

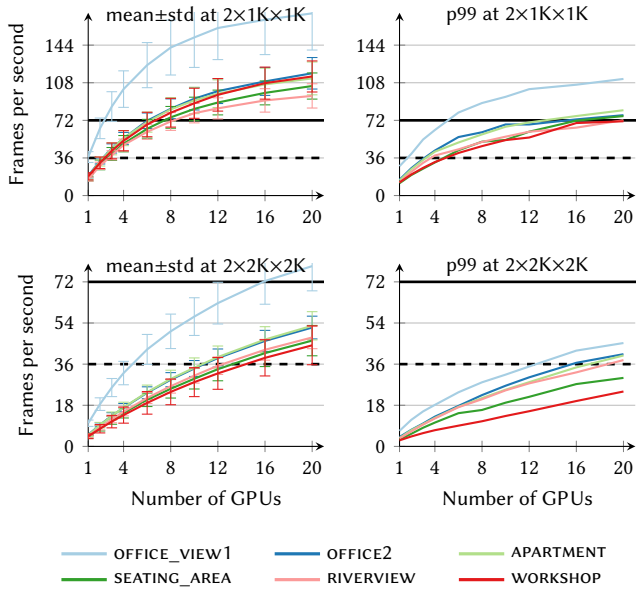


Figure 5: Runtime performance at half (top) and full (bottom) VR resolution (for a Meta Quest Pro) over a prerecorded camera trajectory. Left: Mean and standard deviation of frame rates. Right: The 99th percentile frame time (expressed as FPS) is indicative of the worst-case frame rate.

due to kernel launches, the VR compositor, or OpenGL display pipeline) or insufficient parallelism available at lower resolutions.

7 DISCUSSION

Aggressive pruning. Like most pruning approaches, we threshold density for determining if a voxel is occupied or not. For bounded scenes with mostly solid surfaces, more aggressive pruning can be applied with a larger threshold (e.g., $\alpha = 0.3$) and finer grid resolution (e.g., 1024^3), which results in significantly faster rendering (see 'OFFICE_VIEW1' in Figure 5). However, aggressive pruning does not work well for complex real-world scenes, such as reflective surfaces, transparent objects or unbounded scenes. This becomes particularly apparent in VR, where over-pruned areas show box-like artifacts that may not be easily seen in rendered 2D images or videos.

Distance-aware features. Our level-of-detail feature weighting provides our model the flexibility to reproduce distance-dependent appearance such as varying level of detail, or rig shadows. At the same time, we observed that this reduces our model's ability to extrapolate to unseen viewpoints or viewing distances, as feature vectors with unseen weighting may be used at render time. In particular, density can vary depending on distance, which is undesirable. We work around this by pruning as much free-space as possible, so that density cannot suddenly appear when moving through free-space.

8 CONCLUSION

We presented VR-NeRF, the first holistic approach for capture, reconstruction and rendering of high-fidelity walkable spaces in virtual reality. We made several key contributions across all stages of the pipeline to achieve the significantly higher resolution, frame rate and visual fidelity required for comfortable VR viewing of neural radiance fields. We built a one-of-a-kind multi-camera rig that captures thousands of uniformly distributed HDR photos of a scene, integrated a novel perceptual color space for HDR model optimization, devised an efficient feature mip-mapping scheme for level-of-detail rendering, and implemented a multi-GPU renderer that achieves comfortable VR viewing on our demo machine.

ACKNOWLEDGMENTS

We would like to thank Ada Lopaczynski, Autumn Trimble, Gadsden Merrill, Julia Buffalini, Kevyn McPhail, and Shukri Abdul Jalil for their exceptional technical support, and Alexandre Chapiro, Nathan Matsuda, Rafał Mantiuk and Yaser Sheikh for helpful discussions.

REFERENCES

Edward H. Adelson and James R. Bergen. 1991. The Plenoptic Function and the Elements of Early Vision. In *Computational Models of Visual Processing*. 3–20.
Agisoft, LLC. 2023. Metashape 2.0.
Samir Aroudj, Steven Lovegrove, Eddy Ilg, Tanner Schmidt, Michael Goesele, and Richard Newcombe. 2022. ERF: Explicit Radiance Field Reconstruction From Scratch. (2022). arXiv:2203.00051.
Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. 2021. Mip-NeRF: A Multiscale Representation for Anti-Aliasing Neural Radiance Fields. In *ICCV*. DOI: 10.1109/ICCV48922.2021.00580
Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. 2022. Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields. In *CVPR*. DOI: 10.1109/CVPR52688.2022.00539
Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. 2023. Zip-NeRF: Anti-Aliased Grid-Based Neural Radiance Fields. In *ICCV*.
Dean Beeler, Ed Hutchins, and Paul Pedriana. 2016. Asynchronous Spacewarp. https://developer.oculus.com/blog/asynchronous-spacewarp/. Oculus Developer Blog.
Tobias Bertel, Mingze Yuan, Reuben Lindroos, and Christian Richardt. 2020. OmniPhotos: Casual 360° VR Photography. *ACM Trans. Graph.* 39, 6 (2020), 267:1–12. DOI: 10.1145/3414685.3417770
Michael Broxton, John Flynn, Ryan Overbeck, Daniel Erickson, Peter Hedman, Matthew DuVall, Jason Dourgarian, Jay Busch, Matt Whalen, and Paul Debevec. 2020. Immersive Light Field Video with a Layered Mesh Representation. *ACM Trans. Graph.* 39, 4 (2020), 86:1–15. DOI: 10.1145/3386569.3392485
Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. 2022. TensoRF: Tensorial Radiance Fields. In *ECCV*. DOI: 10.1007/978-3-031-19824-3_20
Abe Davis, Marc Levoy, and Frédo Durand. 2012. Unstructured Light Fields. *Comput. Graph. Forum* 31, 2 (2012), 305–314. DOI: 10.1111/j.1467-8659.2012.03009.x
John Flynn, Michael Broxton, Paul Debevec, Matthew DuVall, Graham Fyffe, Ryan Overbeck, Noah Snavely, and Richard Tucker. 2019. DeepView: View Synthesis With Learned Gradient Descent. In *CVPR*. 2367–2376. DOI: 10.1109/CVPR.2019.00247
Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. 1996. The lumigraph. In *SIGGRAPH*. 43–54. DOI: 10.1145/237170.237200
Param Hanji, Fangcheng Zhong, and Rafał K. Mantiuk. 2020. Noise-Aware Merging of High Dynamic Range Image Stacks without Camera Calibration. In *ECCV Workshops*. DOI: 10.1007/978-3-030-67070-2_23
Peter Hedman, Tobias Ritschel, George Drettakis, and Gabriel Brostow. 2016. Scalable Inside-Out Image-Based Rendering. *ACM Trans. Graph.* 35, 6 (2016), 231:1–11. DOI: 10.1145/2980179.2982420
Xin Huang, Qi Zhang, Ying Feng, Hongdong Li, Xuan Wang, and Qing Wang. 2022. HDR-NeRF: High Dynamic Range Neural Radiance Fields. In *CVPR*. DOI: 10.1109/CVPR52688.2022.01785
Hyeonjoong Jang, Andréas Meuleman, Dahyun Kang, Donggun Kim, Christian Richardt, and Min H. Kim. 2022. Egocentric Scene Reconstruction from an Omnidirectional Video. *ACM Trans. Graph.* 41, 4 (2022), 100:1–12. DOI: 10.1145/3528223.3530074
Yifan Jiang, Peter Hedman, Ben Mildenhall, Dejia Xu, Jonathan T. Barron, Zhangyang Wang, and Tianfan Xue. 2023. AligNeRF: High-Fidelity Neural Radiance Fields via Alignment-Aware Training. In *CVPR*. DOI: 10.1109/CVPR52729.2023.00013
Hanbyul Joo, Tomas Simon, Xulong Li, Hao Liu, Lei Tan, Lin Gui, Sean Banerjee, Timothy Godisart, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. 2019. Panoptic Studio: A Massively Multiview System for Social Interaction Capture. *TPAMI* 41, 1 (2019), 190–204. DOI: 10.1109/TPAMI.2017.2782743
Kim Jun-Seong, Kim Yu-Ji, Moon Ye-Bin, and Tae-Hyun Oh. 2022. HDR-Plenoxels: Self-Calibrating High Dynamic Range Radiance Fields. In *ECCV*. DOI: 10.1007/978-3-031-19824-3_23

Takeo Kanade, P. J. Narayanan, and Peter W. Rander. 1995. Virtualized reality: concepts and early results. In *ICCV Workshops*. 69–76. DOI: 10.1109/WVRS.1995.476854

Sing Bing Kang and Richard Weiss. 2000. Can We Calibrate a Camera Using an Image of a Flat, Textureless Lambertian Surface?. In *ECCV*. DOI: 10.1007/3-540-45053-X_41

Changil Kim, Henning Zimmer, Yael Pritch, Alexander Sorkine-Hornung, and Markus Gross. 2013. Scene reconstruction from high spatio-angular resolution light fields. *ACM Trans. Graph.* 32, 4 (2013), 73:1–12. DOI: 10.1145/2461912.2461926

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*.

Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. 2017. Tanks and Temples: Benchmarking Large-scale Scene Reconstruction. *ACM Trans. Graph.* 36, 4 (2017), 78:1–13. DOI: 10.1145/3072959.3073599

Timo Kunkel. 2022. The Perceptual Quantizer: Design Considerations and Applications. (2022). Talk at ICC HDR Experts Day.

Marc Levoy and Pat Hanrahan. 1996. Light field rendering. In *SIGGRAPH*. 31–42. DOI: 10.1145/237170.237199

Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H. Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-Hsuan Lin. 2023. Neuralangelo: High-Fidelity Neural Surface Reconstruction. In *CVPR*. 8456–8465. DOI: 10.1109/CVPR52729.2023.00817

Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. 2021. BARF: Bundle-Adjusting Neural Radiance Fields. In *ICCV*. DOI: 10.1109/ICCV48922.2021.00569

Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. 2020. Neural Sparse Voxel Fields. In *NeurIPS*.

Siwei Lyu. 2010. Estimating Vignetting Function from a Single Image for Image Authentication. In *ACM Workshop on Multimedia and Security*. 3–12. DOI: 10.1145/1854229.1854233

Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. 2021. NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In *CVPR*. DOI: 10.1109/CVPR46437.2021.00713

Andreas Meuleman, Yu-Lun Liu, Chen Gao, Jia-Bin Huang, Changil Kim, Min H. Kim, and Johannes Kopf. 2023. Progressively Optimized Local Radiance Fields for Robust View Synthesis. In *CVPR*. 16539–16548. DOI: 10.1109/CVPR52729.2023.01587

Ben Mildenhall, Peter Hedman, Ricardo Martin-Brualla, Pratul Srinivasan, and Jonathan T. Barron. 2022. NeRF in the Dark: High Dynamic Range View Synthesis from Noisy Raw Images. In *CVPR*. DOI: 10.1109/CVPR52688.2022.01571

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *ECCV*. DOI: 10.1007/978-3-030-58452-8_24

Scott Miller, Mahdi Nezamabadi, and Scott Daly. 2013. Perceptual Signal Coding for More Efficient Usage of Bit Codes. *SMPTE Motion Imaging Journal* 122, 4 (2013), 52–59. DOI: 10.5594/j18290

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. 2022. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *ACM Trans. Graph.* 41, 4 (2022), 102:1–15. DOI: 10.1145/3528223.3530127

Sergio Orts-Escolano, Christoph Rhemann, Sean Fanello, Wayne Chang, Adarsh Kowdle, Yury Degtyarev, David Kim, Philip L. Davidson, Sameh Khamis, Mingsong Dou, Vladimir Tankovich, Charles Loop, Qin Cai, Philip A. Chou, Sarah Mennicken, Julien Valentin, Vivek Pradeep, Shenlong Wang, Sing Bing Kang, Pushmeet Kohli, Yuliya Lutchyn, Cem Keskin, and Shahram Izadi. 2016. Holoportation: Virtual 3D Teleportation in Real-time. In *UIST*. 741–754. DOI: 10.1145/2984511.2984517

Jin-Si R. Over, Andrew C. Ritchie, Christine J. Kranenburg, Jenna A. Brown, Daniel D. Buscombe, Tom Noble, Christopher R. Sherwood, Jonathan A. Warrick, and Phillipe A. Wernette. 2021. *Processing coastal imagery with Agisoft Metashape Professional Edition, version 1.6—Structure from motion workflow documentation*. Open-File Report 2021-1039. U.S. Geological Survey. DOI: 10.3133/ofr20211039

Ryan Styles Overbeck, Daniel Erickson, Daniel Evangelakos, Matt Pharr, and Paul Debevec. 2018. A System for Acquiring, Compressing, and Rendering Panoramic Light Field Stills for Virtual Reality. *ACM Trans. Graph.* 37, 6 (2018), 197:1–15. DOI: 10.1145/3272127.3275031

Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Steven M. Seitz, and Ricardo-Martin Brualla. 2021. Nerfies: Deformable Neural Radiance Fields. In *ICCV*. DOI: 10.1109/ICCV48922.2021.00581

Albert Parra Pozo, Michael Toksvig, Terry Filiba Schrager, Joyse Hsu, Uday Mathur, Alexander Sorkine-Hornung, Rick Szeliski, and Brian Cabral. 2019. An Integrated 6DoF Video Camera and System Design. *ACM Trans. Graph.* 38, 6 (2019), 216:1–16. DOI: 10.1145/3355089.3356555

Julien Philip, Sébastien Morgenthaler, Michaël Gharbi, and George Drettakis. 2021. Free-viewpoint Indoor Neural Relighting from Multi-view Stereo. *ACM Trans. Graph.* 40, 5 (2021), 194:1–18. DOI: 10.1145/3469842

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. 2019. On the Spectral Bias of Neural Networks. In *ICML*. 5301–5310.

Erik Reinhard, Greg Ward, Sumanta Pattanaik, and Paul Debevec. 2006. *High Dynamic Range Imaging – Acquisition, Display and Image-Based Lighting*. Morgan Kaufmann.

Christian Richardt, James Tompkin, and Gordon Wetzstein. 2020. Capture, Reconstruction, and Representation of the Visual Real World for Virtual Reality. In *Real VR – Immersive Digital Reality: How to Import the Real World into Head-Mounted Immersive Displays*. 3–32. DOI: 10.1007/978-3-030-41816-8_1

Radu Alexandru Rosu and Sven Behnke. 2023. PermutoSDF: Fast Multi-View Reconstruction with Implicit Surfaces using Permutohedral Lattices. In *CVPR*. DOI: 10.1109/CVPR52729.2023.00818

Darius Rückert, Linus Franke, and Marc Stamminger. 2022. ADOP: Approximate Differentiable One-Pixel Point Rendering. *ACM Trans. Graph.* 41, 4 (2022), 99:1–14. DOI: 10.1145/3528223.3530122

SMPTE. 2014. *High Dynamic Range Electro-Optical Transfer Function of Mastering Reference Displays*. SMPTE Standard ST 2084:2014. Society of Motion Picture and Television Engineers. DOI: 10.5594/SMPTE.ST2084.2014

Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. 2019. The Replica Dataset: A Digital Replica of Indoor Spaces. (2019). arXiv:1906.05797.

Cheng Sun, Min Sun, and Hwann-Tzong Chen. 2022. Direct Voxel Grid Optimization: Super-fast Convergence for Radiance Fields Reconstruction. In *CVPR*. DOI: 10.1109/CVPR52688.2022.00538

Towaki Takikawa, Joey Litalien, Kangxue Yin, Karsten Kreis, Charles Loop, Derek Nowrouzezahrai, Alec Jacobson, Morgan McGuire, and Sanja Fidler. 2021. Neural Geometric Level of Detail: Real-Time Rendering With Implicit 3D Shapes. In *CVPR*. 11358–11367. DOI: 10.1109/CVPR46437.2021.01120

Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P. Srinivasan, Jonathan T. Barron, and Henrik Kretzschmar. 2022. Block-NeRF: Scalable Large Scene Neural View Synthesis. In *CVPR*. DOI: 10.1109/CVPR52688.2022.00807

Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, David Mcallister, Justin Kerr, and Angjoo Kanazawa. 2023. Nerfstudio: A Modular Framework for Neural Radiance Field Development. In *SIGGRAPH Conference Proceedings*. 72:1–12. DOI: 10.1145/3588432.3591516

Ayush Tewari, Justus Thies, Ben Mildenhall, Pratul Srinivasan, Edgar Tretschk, Yifan Wang, Christoph Lassner, Vincent Sitzmann, Ricardo Martin-Brualla, Stephen Lombardi, Tomas Simon, Christian Theobalt, Matthias Niessner, Jonathan T. Barron, Gordon Wetzstein, Michael Zollhöfer, and Vladislav Golyanik. 2022. Advances in Neural Rendering. *Comput. Graph. Forum* 41, 2 (2022), 703–735. DOI: 10.1111/cgf.14507

Haithem Turki, Deva Ramanan, and Mahadev Satyanarayanan. 2022. Mega-NeRF: Scalable Construction of Large-Scale NeRFs for Virtual Fly-Throughs. In *CVPR*. 12922–12931. DOI: 10.1109/CVPR52688.2022.01258

Ziyu Wan, Christian Richardt, Aljaž Božič, Chao Li, Vijay Rengarajan, Seonghyeon Nam, Xiaoyu Xiang, Tuotuo Li, Bo Zhu, Rakesh Ranjan, and Jing Liao. 2023. Learning Neural Duplex Radiance Fields for Real-Time View Synthesis. In *CVPR*. DOI: 10.1109/CVPR52729.2023.00803

Peng Wang, Yuan Liu, Zhaoxi Chen, Lingjie Liu, Ziwei Liu, Taku Komura, Christian Theobalt, and Wenping Wang. 2023. F^2-NeRF: Fast Neural Radiance Field Training with Free Camera Trajectories. In *CVPR*. DOI: 10.1109/CVPR52729.2023.00404

Zhongshu Wang, Lingzhi Li, Zhen Shen, Li Shen, and Liefeng Bo. 2022. 4K-NeRF: High Fidelity Neural Radiance Fields at Ultra High Resolutions. (2022). arXiv:2212.04701.

Thomas Whelan, Michael Goesele, Steven J. Lovegrove, Julian Straub, Simon Green, Richard Szeliski, Steven Butterfield, Shobhit Verma, and Richard Newcombe. 2018. Reconstructing Scenes with Mirror and Glass Surfaces. *ACM Trans. Graph.* 37, 4 (2018), 102:1–11. DOI: 10.1145/3197517.3201319

Bennett Wilburn, Neel Joshi, Vaibhav Vaish, Eino-Ville Talvala, Emilio Antunez, Adam Barth, Andrew Adams, Mark Horowitz, and Marc Levoy. 2005. High performance imaging using large camera arrays. *ACM Trans. Graph.* 24, 3 (2005), 765–776. DOI: 10.1145/1073204.1073259

Lance Williams. 1983. Pyramidal Parametrics. *Computer Graphics (Proceedings of SIGGRAPH)* 17, 3 (1983), 1–11. DOI: 10.1145/800059.801126

Tianhao Wu, Fangcheng Zhong, Andrea Tagliasacchi, Forrester Cole, and Cengiz Oztireli. 2022b. D^2NeRF: Self-Supervised Decoupling of Dynamic and Static Objects from a Monocular Video. In *NeurIPS*.

Xiuchao Wu, Jiamin Xu, Zihan Zhu, Hujun Bao, Qixing Huang, James Tompkin, and Weiwei Xu. 2022a. Scalable Neural Indoor Scene Rendering. *ACM Trans. Graph.* 41, 4 (2022), 98:1–16. DOI: 10.1145/3528223.3530153

Wenqi Xian, Aljaž Božič, Noah Snavely, and Christoph Lassner. 2023. Neural Lens Modeling. In *CVPR*. 8435–8445. DOI: 10.1109/CVPR52729.2023.00815

Yuanbo Xiangli, Linning Xu, Xingang Pan, Nanxuan Zhao, Anyi Rao, Christian Theobalt, Bo Dai, and Dahua Lin. 2022. BungeeNeRF: Progressive Neural Radiance Field for Extreme Multi-scale Scene Rendering. In *ECCV*. DOI: 10.1007/978-3-031-19824-3_7

Jiamin Xu, Xiuchao Wu, Zihan Zhu, Qixing Huang, Yin Yang, Hujun Bao, and Weiwei Xu. 2021. Scalable Image-Based Indoor Scene Rendering with Reflections. *ACM Trans. Graph.* 40, 4 (2021), 60:1–14. DOI: 10.1145/3450626.3459849

Linning Xu, Yuanbo Xiangli, Sida Peng, Xingang Pan, Nanxuan Zhao, Christian Theobalt, Bo Dai, and Dahua Lin. 2023. Grid-guided Neural Radiance Fields for

Large Urban Scenes. In *CVPR*. DOI: 10.1109/CVPR52729.2023.00802

Jiawei Yang, Marco Pavone, and Yue Wang. 2023. FreeNeRF: Improving Few-shot Neural Rendering with Free Frequency Regularization. In *CVPR*. DOI: 10.1109/CVPR52729.2023.00798

Jae Shin Yoon, Kihwan Kim, Orazio Gallo, Hyun Soo Park, and Jan Kautz. 2020. Novel View Synthesis of Dynamic Scenes with Globally Coherent Depths from a Monocular Camera. In *CVPR*. DOI: 10.1109/CVPR42600.2020.00538

Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. 2022. MonoSDF: Exploring Monocular Geometric Cues for Neural Implicit Surface Reconstruction. In *NeurIPS*.

Yuqi Zhang, Guanying Chen, and Shuguang Cui. 2023. GP-NeRF: Efficient Large-scale Scene Representation with a Hybrid of High-resolution Grid and Plane Features. (2023). arXiv:2303.03003.

Jingsen Zhu, Yuchi Huo, Qi Ye, Fujun Luan, Jifan Li, Dianbing Xi, Lisha Wang, Rui Tang, Wei Hua, Hujun Bao, and Rui Wang. 2023. I^2-SDF: Intrinsic Indoor Scene Reconstruction and Editing via Raytracing in Neural SDFs. In *CVPR*. DOI: 10.1109/CVPR52729.2023.01202

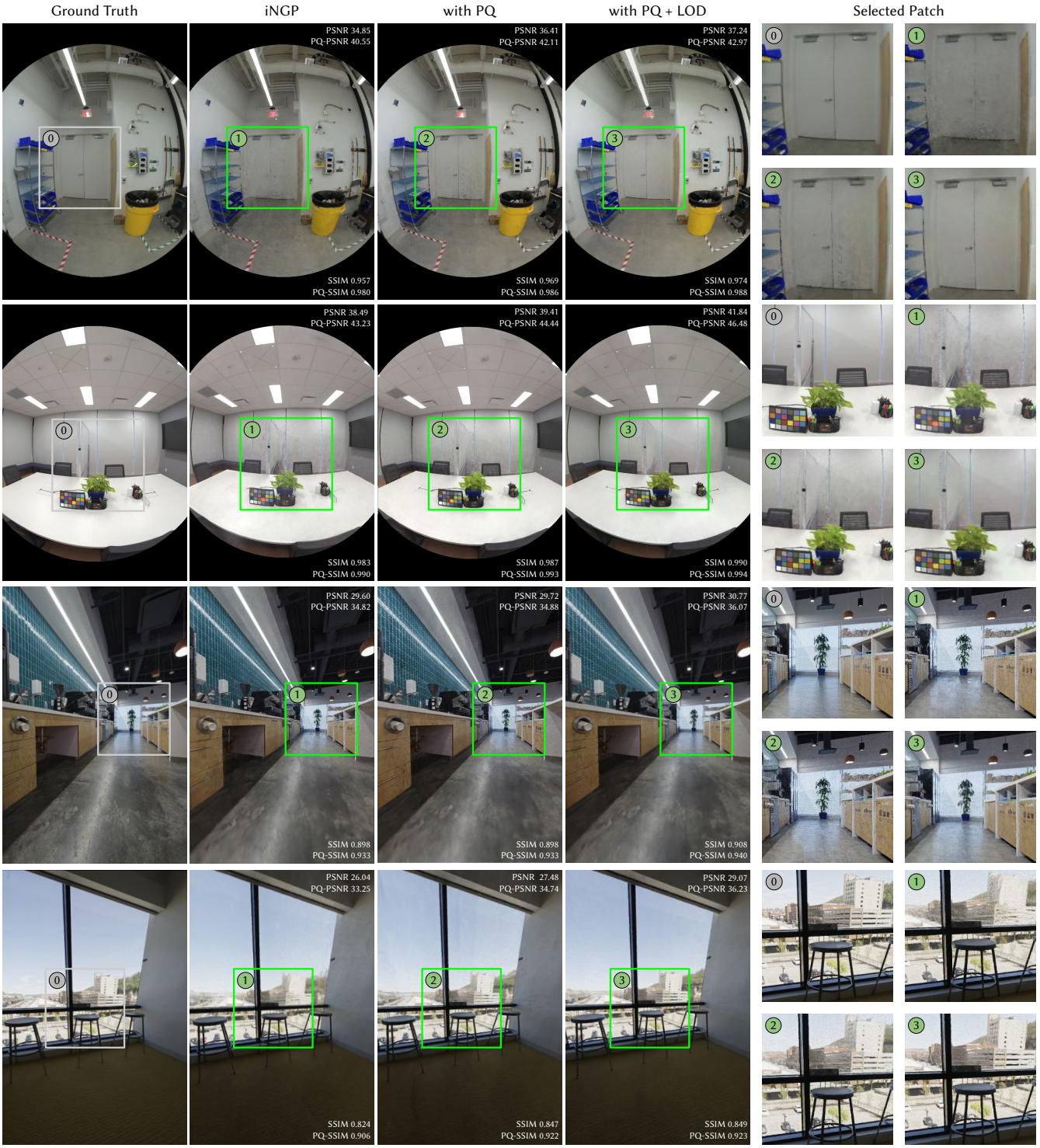


Figure 6: Qualitative comparisons between (1) iNGP, (2) iNGP with PQ color space, and (3) iNGP with PQ color space and LOD. The four selected examples show the improvements over the baselines by adding PQ color space and LOD in combination, which leads to cleaner appearance and geometry finer details. The use of PQ color space stablizes the learning of correct radiance values, while the inclusion of LOD helps to learn a cleaner appearance and geometry that is robust to distance-dependent appearance variations. By dynamically allocating model capacity to the sampled points based on the needed level of detail, it further reveals more details over the ablated counterparts. (Images are white-balanced and tonemapped for better visualization.)

Figure 7: Sweep of exposure values. For scenes with high dynamic range (e.g., bright outdoor views in the RIVERVIEW scene), one can freely adjust the exposure setting at render time by manipulating the tonemapping from PQ color space to sRGB space.

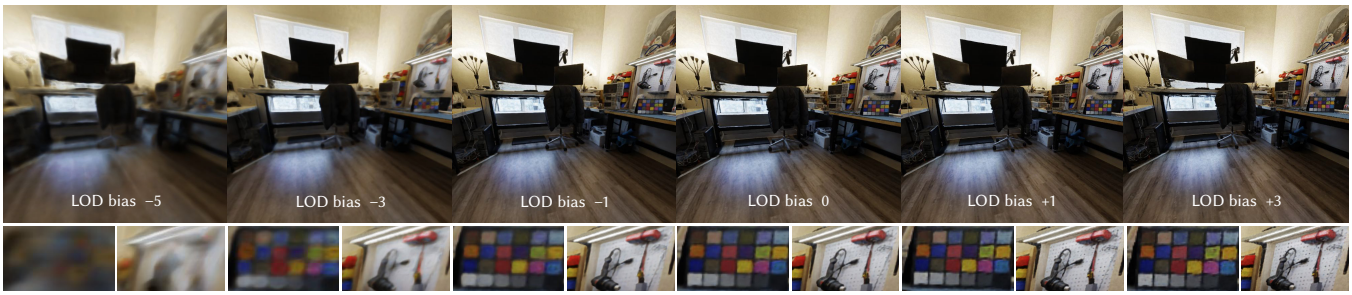


Figure 8: Sweep of LOD bias. We interpolate between a negative LOD bias of −5 and a positive LOD bias of +3 applied on top of the original estimated LOD value for each sample point on the APARTMENT scene. A negative LOD bias blurs the rendering by masking out grid features representing high-frequency details, while a positive LOD bias helps reveal sharper details.

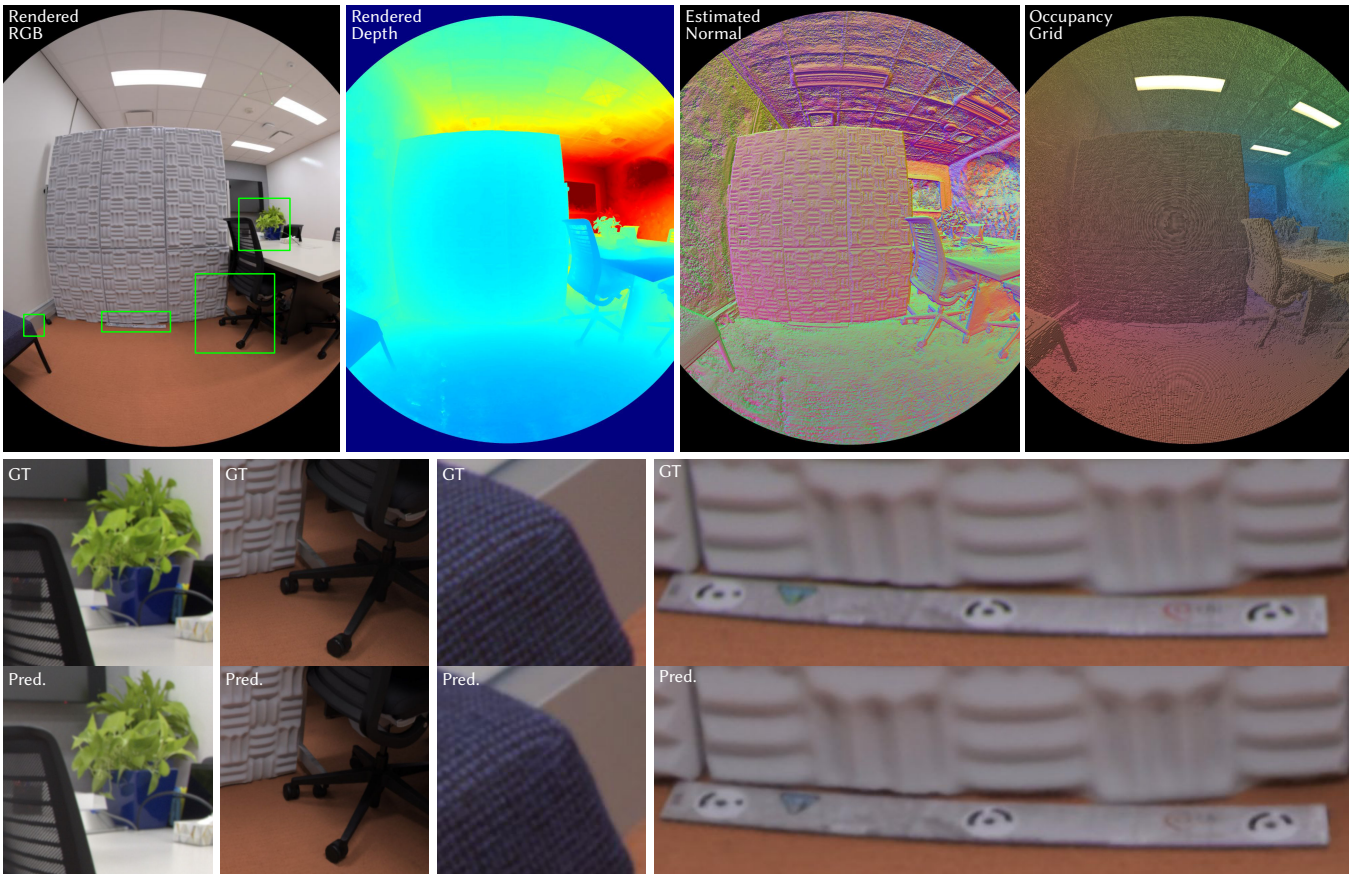


Figure 9: OFFICE2 results rendered at 4K resolution, trained with 400K iterations. Top row: from left to right, we show (1) rendered RGB image, (2) estimated depth map, (3) estimated normals, and (4) the occupancy grid. Bottom row: The highlighted patches reveal sufficient fine details.

A ADDITIONAL CAPTURE DETAILS

The system and procedure described in Section 3 is the third iteration of our capture stack. The first version of the "rig" was a single handheld DSLR camera, but achieving the desired capture density quickly became tedious. This was the primary motivation for the design of the second version of the rig, v1, which featured 9 cameras. While a substantial improvement, the low camera count on v1 still required captures with the rig facing multiple directions. Fortunately, v1 was designed with upgradability in mind, and we were quickly able to add an additional 13 cameras. The result is the current configuration, v2, which again simplified the capture procedure and further increased our data quality.

A.1 "Rig" v0 — Handheld DSLR

Capture hardware. We began by performing handheld captures with a single Canon 1D X Mark II camera, which can take 20 megapixel photos. This was paired with a Canon EF 8–15mm f/4L Fisheye USM lens, set to 8 mm focal length to ensure the highest possible field of view, which minimized the number of images required to achieve high viewing direction coverage. A cellphone camera was initially considered, but ultimately rejected due to the lack of interchangeable lens and insufficient field of view on the existing lenses. We also experimented with using a tripod for additional stability during capture, but found it too cumbersome to continuously reposition it and thus removed it.

Capture procedure. Each capture began by picking a reasonable ISO, shutter speed, and f-stop which would be fixed for the scene – typically somewhere around ISO 1000, 1/40 seconds, and $f/4$. The lens was set to its widest setting, at 8 mm. We then walked multiple loops around the scene, taking an image at small steps, typically about 30 cm, along each one. The camera would be held level horizontally, and its height would be increased each loop, starting at approximately knee height and increasing in 20–50 cm intervals until the camera was above the head height of the person performing the capture. This typically resulted in approximately 100–200 photos being captured for a single scene in 1–2 hours.

A.2 Eyeful Tower v1 — 9 fisheye cameras

The captures we performed with the single handheld camera were enough to get us started, but also clearly had some limitations. The biggest, and most obvious, was the difficulty and time required to ensure sufficient scene coverage. We also observed over- and under-exposure of different parts of our scenes, such as when looking through windows (at the sun) or at shadows. We endeavored to address both of these issues with a rig that featured multiple rigidly mounted cameras, at differing heights, which could capture simultaneously. Cameras at different heights would allow captures similar to the handheld setup with only a single pass of the scene, rather than one per height as before. The rigid mounting would also enable an exposure bracket to be taken, enabling HDR images to be generated.

Camera, lens, and exposure. The construction of a multi-camera rig gave us the opportunity to take a closer look at our camera and lens selection. After carefully considering options for both professional and machine vision cameras, we chose the Sony $\alpha 1$ mirrorless interchangeable lens cameras for striking a good balance between resolution (50 megapixels), dynamic range (14 stops), ease-of-use (available analog triggers and Ethernet socket), as well as form factor (smaller than our previous Canon DSLR camera). We continued to use the same lens as before, via a Metabones EF-to-E-mount adapter, but this time zoomed to 12 mm to fill more of the camera's sensor, as shown in Figure 11. We use ISO 500, the camera's higher native ISO, for minimal imaging noise, and set the aperture to $f/8$ with a focus distance of 1 m for a large depth of field. The cameras are configured to take a 9-image exposure bracket, 1 stop apart (−4 EV to +4 EV), as shown in Figure 13. The center exposure value is adjusted per-scene and is typically between 1/200 and 1/60 of a second. RAW and JPEG images are stored redundantly on the two SD cards in each camera.

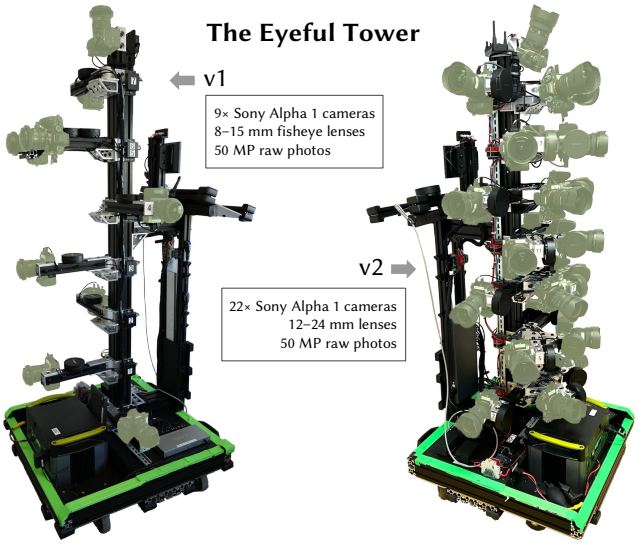


Figure 10: The two versions of the Eyeful Tower camera rig.

Mechanical design and camera placement. We designed the capture rig using 80/20 extruded aluminium around an 80×80 cm base with a 1.8 m vertical pole, allowing for substantial adjustability and expandability. The pole held 7 camera brackets, each capable of supporting a single camera, whose height could be adjusted, and whose direction could be adjusted within 180 degrees horizontally. We positioned the cameras left, forward, and right in an alternating fashion, as shown on the left side of Figure 10, attempting to maximize scene coverage from a single rig position while allowing sufficient space for the operator to not be visible in camera images when standing behind the rig. One forward-facing bracket, at roughly eye height, was specially modified to support a second camera that would be held out for validation. A final camera was added at the top for ceiling coverage, which was otherwise not present, for a total of 9 cameras on v1.

Electrical design. At the base of the rig is a 1.5 kWh Li-ion battery, capable of supplying regulated 12 V DC and 120 V AC power via an integrated inverter. The 12 V bus is used to provide power to all cameras and a Raspberry Pi 4 via a 12 V to USB-PD adapter.

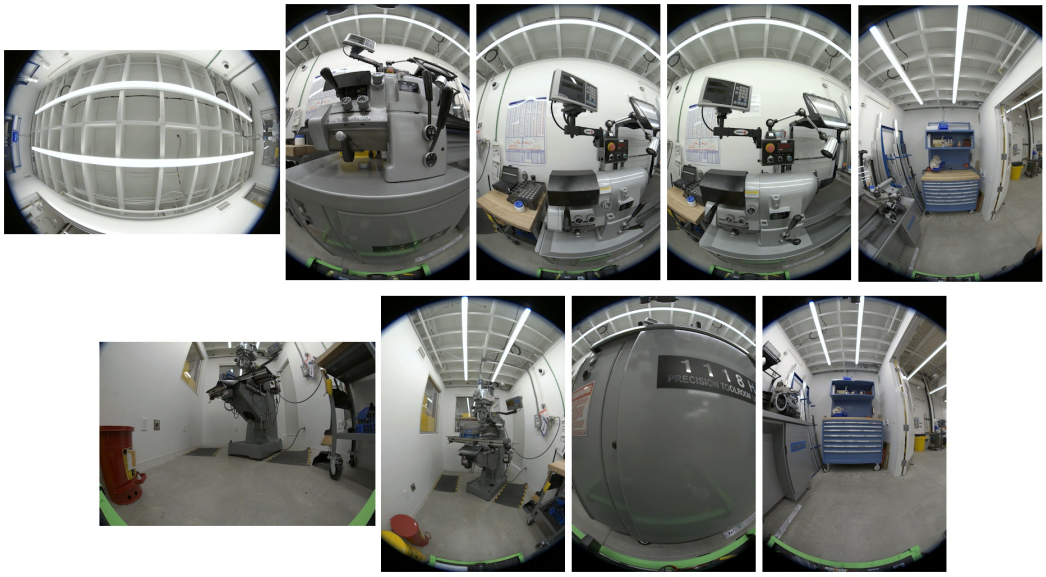


Figure 11: Example frames from the 'WORKSHOP' dataset captured using the 9 cameras in Eyeful Tower v1.

Figure 12: Example frames from the 'APARTMENT' dataset captured using the 22 cameras in Eyeful Tower v2.

1/3200 s (−6 stops) 1/1600 s (−5 stops) 1/800 s (−4 stops) 1/400 s (−3 stops) 1/200 s (−2 stops) 1/100 s (−1 stops) 1/50 s (±0 stops) 1/25 s (+1 stops) 1/13 s (+2 stops)

Figure 13: To capture the high dynamic range (HDR) of the real world, we take nine exposures at increasing shutter speeds, and merge these photos into a single HDR image. We can therefore reproduce the full dynamic range of input exposures, including the brightest and darkest regions.

The cameras are powered by internal Li-ion batteries, but are continually trickle charged via USB. The 120 V AC output powers a 24-port 1-Gigabit Ethernet switch that connects all the cameras to the Raspberry Pi. The Pi runs custom capture software to enable formatting of all camera SD cards, camera parameter updates, and simultaneous bracket triggering via the Sony Camera Remote SDK. The switch also offers a 10GbE SFP+ port which is used to offload data from the cameras to a PC for downstream processing. The battery is able to power the entire system for 6 to 8 hours of use, and can be recharged via normal 120 V wall power overnight.

Capture procedure. Cameras at multiple heights and viewing directions simplifies the coverage problem during capture from 6D to 3D, as we now only need to tile the floor with the rig facing a few directions. The general capture strategy is described in Section 3.2.2, with example camera positions shown in Figure 14. For this initial version of the rig, however, we needed to capture in four separate orientations (facing forward, backward, left, and right, versus just forward and backward as described above), in order to ensure 360 horizontal degrees of coverage for each rig position at each height.

A.3 Eyeful Tower v2 — 22 rectilinear lenses

The additional cameras of Eyeful Tower v1 offered a substantial usability and data quality improvement over the single handheld camera, but the sparse positioning still necessitated four passes of the scene. The current version of the rig, v2, attempts to address this by adding yet more cameras, increasing the total count to 22, as shown on the right in Figure 10. The lenses have also been replaced with rectilinear Sony FE 12–24mm F/2.8 GM lenses, selected for their higher sharpness (2–3× MTF50) and lower chromatic aberration (less than half, in pixels). The camera and lens parameters are kept the same as before — 12 mm zoom, ISO 500, $f/8$ aperture, and 1 m focus distance. An example set of captured images is shown in Figure 12. The additional cameras more than compensate for the slightly lower per-camera FOV, and enable us to capture a scene in two passes (rig facing forward and backward) rather than the four that v1 required.

B MODEL IMPLEMENTATION DETAILS

B.1 Vignetting and Lens Distortion

Vignetting effects are commonly present in wide-angle lenses, such as the fisheye and wide-angle lenses used in our capture system (see Appendix A). We parameterize the vignetting effect for each camera using the Kang and Weiss model [2000] with $I' \approx (1 - \alpha r)I$, where we make α and the principal point (c_x, c_y) used for computing r learnable parameters for each camera sensor. This allows the model to fit to the radial falloffs. Without modeling vignetting [Lyu 2010] explicitly, the model is likely to overfit on training views by casting unwanted black floats everywhere in the air to accounting for the brightness decrease towards the edge of each image frame. Figure 15 shows that modeling vignetting explicitly is crucial for certain subset of data. In scenarios where the test camera has unlearned vignetting parameters, one can optionally optimize the vignetting model for the test camera lens before inference with all the other parameters fixed. The optimization can be done quickly in hundreds or thousands iterations. Note that we do not optimise for lens distortions [Xian et al. 2023] in our current implementation, which is left as future work to further improve pixel-wise alignment and accuracy.

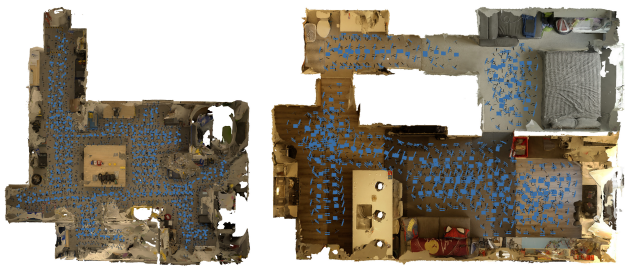


Figure 14: Visualized camera positions for the WORKSHOP (left) and APARTMENT (right) scenes.

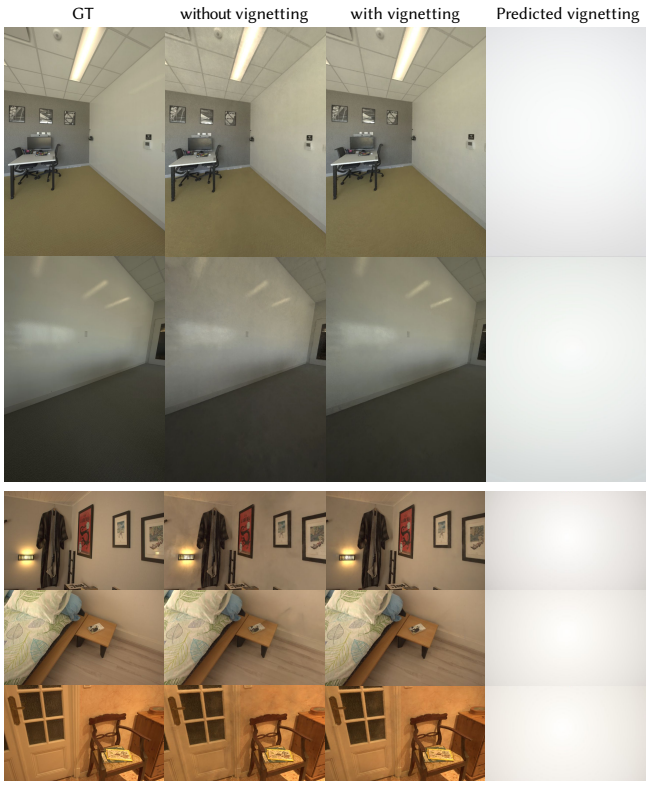


Figure 15: Ablation on vignetting effects. Without explicitly modeling vignetting, the model has difficulty in explaining away the inconsistent appearance brought by different camera lenses. We found that this effect is easier to observe in textureless areas, such as walls and floors in OFFICE_VIEW1 (first row) and OFFICE_VIEW2 (second row), with noticeable dark floaters in front of cameras. The vignetting effect is found more strongly in the Inria datasets [Philip et al. 2021], as shown in the bottom figure.

B.2 Loss Design

Image Loss. With PQ color space ranged in [0,1], we can directly use L_1/L_2 for reconstruction loss metric without biasing towards certain color range. We prioritize L_1 loss in our experiments, as it is generally regarded as a more robust loss for outliers and provide sparser solutions compared to L_2 loss. We also experimentally found that L_1 leads to sharper details compared to L_2 loss.

Depth Regularizations. NeRF reconstruction can be challenging for textureless areas, such as a flat white wall or a featureless floors. Furthermore, the challenging reflections and shadows which caused the abrupt change in the brightness can easily lead to incorrect geometry where the view-dependent effects fail to capture the variances. To address this issue, some approaches use additional geometrical information [Yu et al. 2022], such as depth map guidances. As the monocular depth predicted by off-the-shelf models can only be used in relative scale, we instead resort to the reconstructed mesh from Metashape during the pose estimation stage and project it to the training views as pseudo ground truth depth map for supervision, which allows us to perform direct comparison in absolute scale. To avoid the misguidance from unreliable depth map, we use depth loss only for the early stage of training, where the incorrect geometry can get refined with image reconstruction loss only. Figure 16 shows the effects of using depth loss guidance in early training stages.

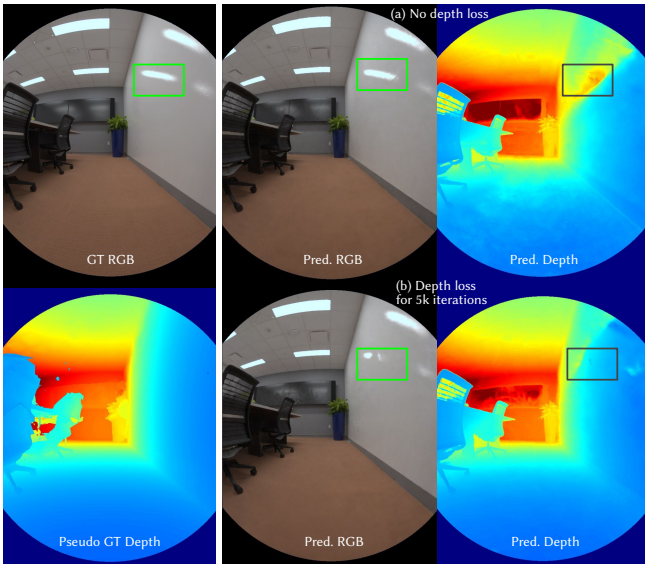


Figure 16: Applying depth loss. The bottom-left image shows the pseudo depth guidance obtained from Metashape's mesh reconstruction. Experimentally, we found that the depth loss converges extremely fast in few hundreds or thousands iterations. The first row shows the results without depth supervision, while the result shown in the second row is supervised by the depth loss for the first 5K iterations. The depth supervision prevents model from cheating reflections with samples placed behind the walls, leading to the inability to recover the highlights.

Distortion Loss. We adopt a simplified version of distortion loss [Barron et al. 2022] that encourages the sparseness of the sample weights along the ray without considering the compactness of sample intervals from the proposal network. In practice, we additionally consider the depth variance loss applied on the inner world (regions unaffected by space warping) that encourages the weights to be concentrated around the estimated depth. The depth variance is calculated among samples cast by a single pixel. Figure 17 shows the effects of applying depth variance loss that learns flat wall without reflections. This trick is suggested to use during the separate pruning stage, where the cleaner geometry is used for obtaining reliable occupancy grid only. This serves as an additional regularization for challenging scenarios such as the highly reflective surfaces, planes with detailed textures, where the depth variance could be relative large leading to incorrect geometry. We optionally add an "empty around camera" loss by placing random samples in the unit sphere around the cameras to avoid near-plane ambiguity, similar to the occlusion loss in FreeNeRF [Yang et al. 2023].

Other Regularizations. Barron et al. [2023] recently proposed to apply a weight-decay loss on the multi-level hash grid features to encourage a normal distribution of learned grid features. We found that this loss can regularize the learning of grid features, leading to more complete geometry and flat surfaces, yet at the cost of lower convergence speed and occasional detail loss.

B.3 Weighted Sampling

Given the large number of rays used to be supervised during training, it is generally impractical to revisit each pixel multiple times

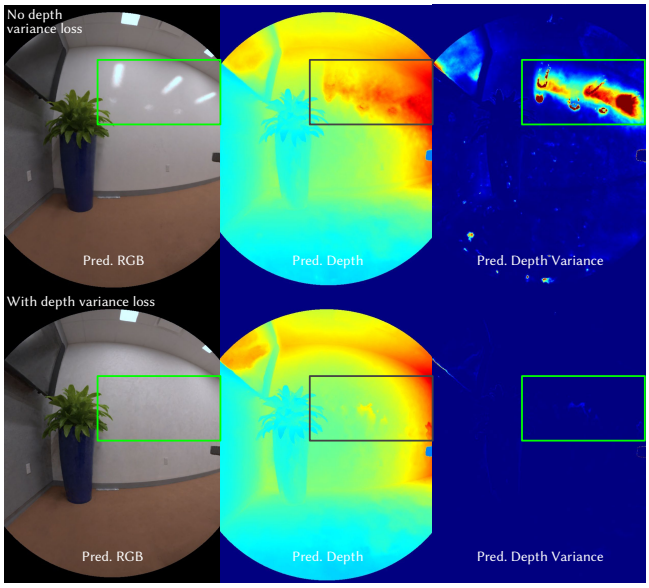


Figure 17: Applying depth variance loss. Depth variance loss can be applied without ground-truth depth supervision. The depth variance map on the top right shows a clear correlation between these reflection regions and the depth variance statistics, which indicates its potential to suppress cheating of appearance changes via wrong geometry.

Table 3: Quantitative comparison results on the Eyeful Tower test set. All results are trained on 1K images for 110K iterations with 1024 samples per ray. We report PSNR/SSIM/LPIPS both in sRGB and PQ color spaces. The best results are highlighted.

Scene	Far-field	iNGP (our implementation)						iNGP with PQ color space						iNGP with PQ color space and LOD					
		PSNR ↑	SSIM ↑	LPIPS ↓	PQ-PSNR↑	PQ-SSIM↑	PQ-LPIPS↓	PSNR ↑	SSIM ↑	LPIPS ↓	PQ-PSNR↑	PQ-SSIM↑	PQ-LPIPS↓	PSNR ↑	SSIM ↑	LPIPS ↓	PQ-PSNR↑	PQ-SSIM↑	PQ-LPIPS↓
APARTMENT	✓	30.68	0.903	0.226	35.16	0.942	0.203	31.06	0.910	0.208	35.52	0.946	0.186	32.36	0.915	0.190	36.61	0.948	0.173
KITCHEN	✓	31.33	0.925	0.216	36.43	0.954	0.166	31.50	0.928	0.210	36.67	0.956	0.161	32.41	0.932	0.184	37.53	0.958	0.146
OFFICE1A		35.71	0.972	0.095	41.75	0.988	0.055	36.20	0.974	0.091	42.43	0.989	0.052	36.82	0.976	0.082	43.05	0.990	0.048
OFFICE1B		27.58	0.880	0.402	33.38	0.952	0.248	28.44	0.895	0.361	34.59	0.960	0.216	29.97	0.914	0.255	36.16	0.966	0.162
OFFICE2		39.84	0.983	0.053	44.46	0.986	0.031	40.26	0.992	0.046	45.12	0.993	0.026	40.71	0.987	0.037	45.53	0.994	0.023
OFFICE_VIEW1	✓	29.75	0.890	0.264	35.14	0.942	0.192	30.20	0.897	0.253	35.63	0.947	0.181	31.80	0.901	0.223	37.09	0.948	0.167
OFFICE_VIEW2	✓	27.19	0.858	0.259	33.14	0.921	0.201	27.64	0.865	0.259	33.73	0.926	0.198	28.08	0.868	0.230	34.09	0.927	0.181
RIVERVIEW	✓	27.17	0.864	0.181	34.09	0.934	0.135	27.83	0.869	0.179	34.77	0.938	0.134	28.47	0.863	0.181	35.51	0.935	0.135
SEATING_AREA	✓	36.85	0.968	0.070	41.93	0.983	0.050	37.16	0.970	0.069	42.26	0.984	0.049	37.86	0.974	0.053	42.92	0.987	0.035
TABLE	✓	34.28	0.952	0.091	40.37	0.974	0.067	34.52	0.954	0.087	40.78	0.975	0.064	35.40	0.962	0.066	41.59	0.980	0.046
WORKSHOP		30.86	0.907	0.155	36.54	0.949	0.115	32.34	0.937	0.102	38.12	0.965	0.074	32.40	0.940	0.101	38.31	0.967	0.073
Average		31.93	0.918	0.183	37.39	0.957	0.133	32.47	0.926	0.170	38.15	0.962	0.122	33.30	0.930	0.146	38.95	0.964	0.108

with limited training time. The training of NeRF is notorious for long training time in order to bring finer details, which usually appear in the later stage of training due to the spectral bias of neural networks [Rahaman et al. 2019]. We also observed from the computed error map that the errors usually dominate in high-frequency details, while areas with relative uniform geometry and appearance have significant lower errors. One can either assign higher loss weights to these pixels representing high-frequency details or sampling them more often during the training. Practically, we compute the Laplacian pyramid for each image during data preparation stage and use them as the indicator for important pixel samples. Instead of using the directly obtained continuous value for pixel weights, we threshold the importance samples by referencing to the 75% percentile of all pixels over the image. See Figure 18 for the construction of weighted map. For perspective images, we additionally consider the effects of wide-angle field-of-view and assigning a sampling weight inverse to the actual radii of each pixel footprint, which better matches the pixel coverage in 3D world. The effectiveness of weighted samples can be seen in Figure 19.

C ADDITIONAL RESULTS AND ANALYSIS

We show a detailed breakdown of our complete quantitative ablation results across all 11 Eyeful Tower datasets in Table 3.

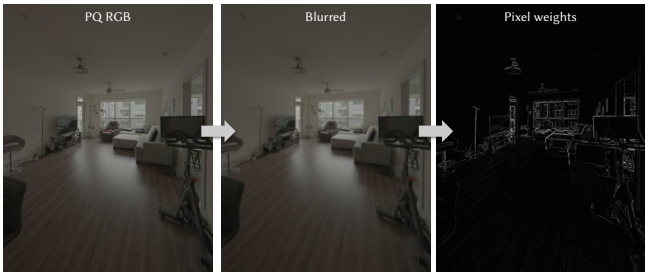


Figure 18: Derivation of pixel weights. We borrow the idea of Laplacian pyramid and derive the importance weight for each pixel to guide sampling.

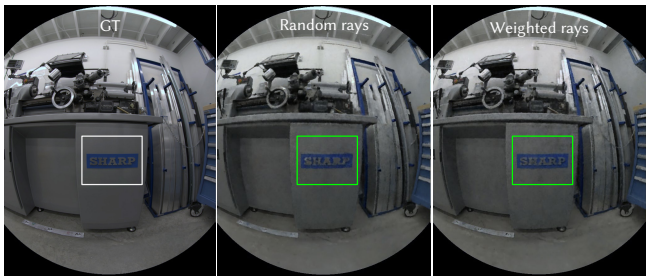


Figure 19: Weighted samples. Effects of revealing finer details and geometry in early training stages (e.g., 2K iterations) with weighted sampling. The sign of 'SHARP' in the scene is quickly revealed with weighted sampling.

C.1 Learning of linear RGB space

Figure 20 visualize the converted PQ color space, where both very bright and dark regions are properly preserved in the converted space. Figure 21 plots the RGB values produced by model linear output layer, activation function, as well as the weighted value used for final volumetric integration during the training process. It is clear to see that directly learning on linear RGB space with safe exponential function may cause the unstable training where the maximum value predicted from model outputs keeps growing and

Figure 20: Illustration of PQ color space. The highlighted regions represent extreme bright/dark areas which are properly handled by the PQ conversion.

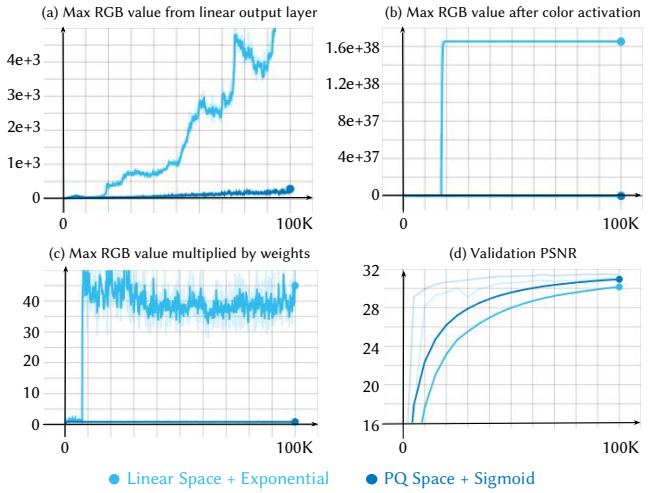


Figure 21: RGB value analysis. These curves show: (a) the maximum RGB value obtained from the linear output layer of the color network, (b) the maximum RGB value after the color activation function, (c) the point-wise color multiplied by the integrated density weight for final color composition, and (d) the validation PSNR showing the rendering fidelity. The two blue lines represent: (1) the baseline using iNGP with linear color space, and exponential activation (light blue), and (2) our PQ color space with sigmoid activation function (dark blue).

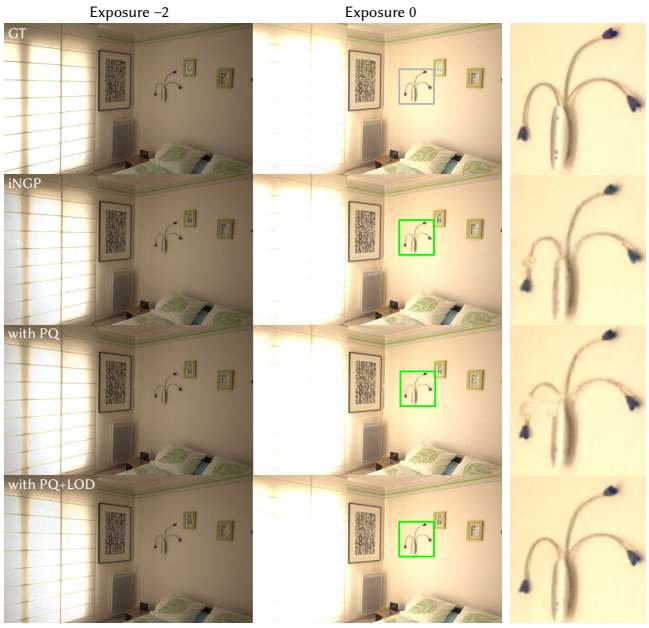


Figure 22: Ablation comparisons on raw Inria dataset [Philip et al. 2021]. The highlighted patches clearly show that iNGP w/ PQ + LOD better preserves the geometry and details compared to the ablated baselines. We also adjust the exposure values to adapt to the bright areas around the window.

fluctuated all the way. In contrast, we need not worry about all these issues in PQ RGB space as we are already work on the bounded PQ space in range [0,1], which gives pretty stabilized learning with commonly used sigmoid activation function.

C.2 Effects of LOD

Figure 22 shows the two ablated modules on Inria dataset. One commonly observed advantage of using LOD feature is its improvements on revealing fine details. We conjecture that by dynamically masking out high-resolution grid features, the model encourages these high-frequency features to only be used for rendering contents with close observations and fine details.

C.3 Ablation on pruning strategy

Figure 24 shows the comparison on different pruning strategies. As 'history pruning' only considers stochastically sampled points visited during the pruning period, it is unlikely to visit all the voxels during the updating period, leading to numerous holes in the obtained occupancy grids. The quality of 'grid pruning' commonly depends on the number of samples placed within each voxel. The estimation accuracy can get improved with increased number of samples yet at the cost of large computation expenses. Furthermore, as these samples are usually evenly places for robustness, it can rarely matches with surface points, leading to box-like artifacts in the obtained geometry. Our joint training combines the merits of each method and achieves accurate pruning results with limited computing budges (4 points for each voxel for grid pruning).

C.4 Mip-NeRF 360 & Inria datasets

We additionally tested on Mip-NeRF 360 dataset [Barron et al. 2022] with our LOD and pruning designs. Figure 25 and Figure 26 show all the 9 scenes used in Barron et al. [2022]. The models are trained at 2K resolution, with properly recovered fine details and accurate occupancy grids. Philip et al. [2021] provide scenes with captured raw images. Figure 27 shows three scenes trained with HDR inputs and PQ color space.

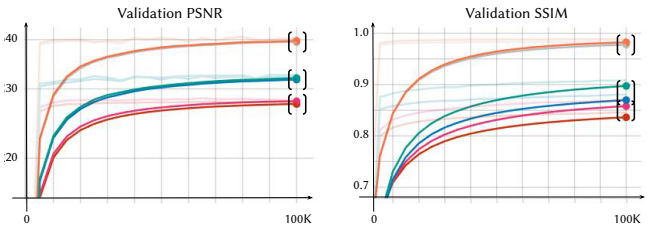


Figure 23: 1K & 2K training. The learning curves show the comparative validation results between training on 1K & 2K resolution datasets. The black brackets include the pair of experiments for a same scene with 1K and 2K version. The quantitative metrics are similar without significant drop when adapting to higher resolution images, which shows the potential of using higher-resolution images for training sufficiently long.

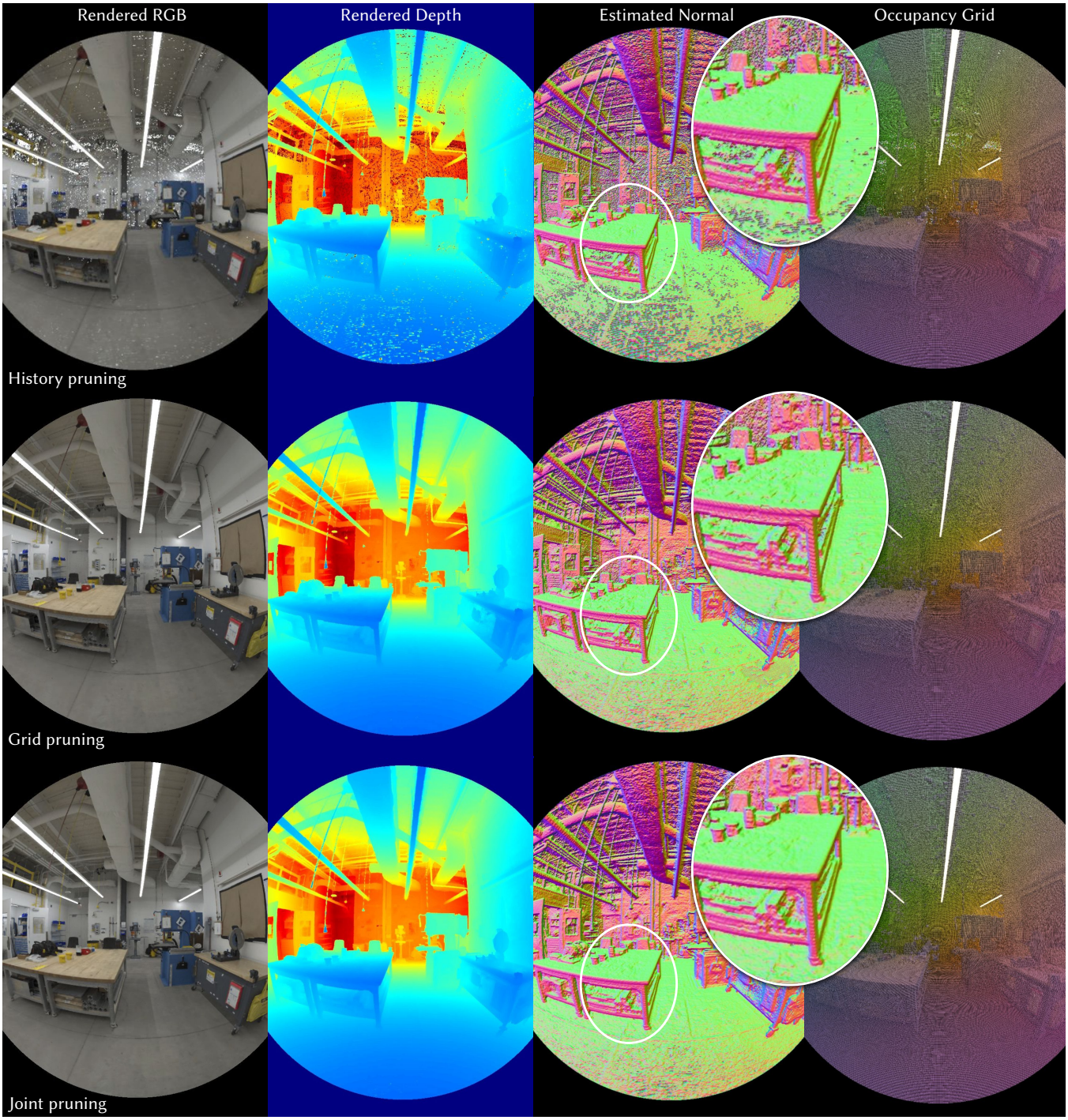


Figure 24: Ablation on pruning strategy. From top to bottom, we compare three alternative pruning strategies. 'Joint pruning' (bottom) leverages the advantage of both 'history pruning' (top) and 'grid pruning' (middle) by placing important sample points observed during training and also densely evaluating voxel grids with sufficient coverage. The obtained occupancy grid shown on the right is clean and accurate, and the derived depth map and normal map indicate the well-preserved geometry compared to each individual strategy.

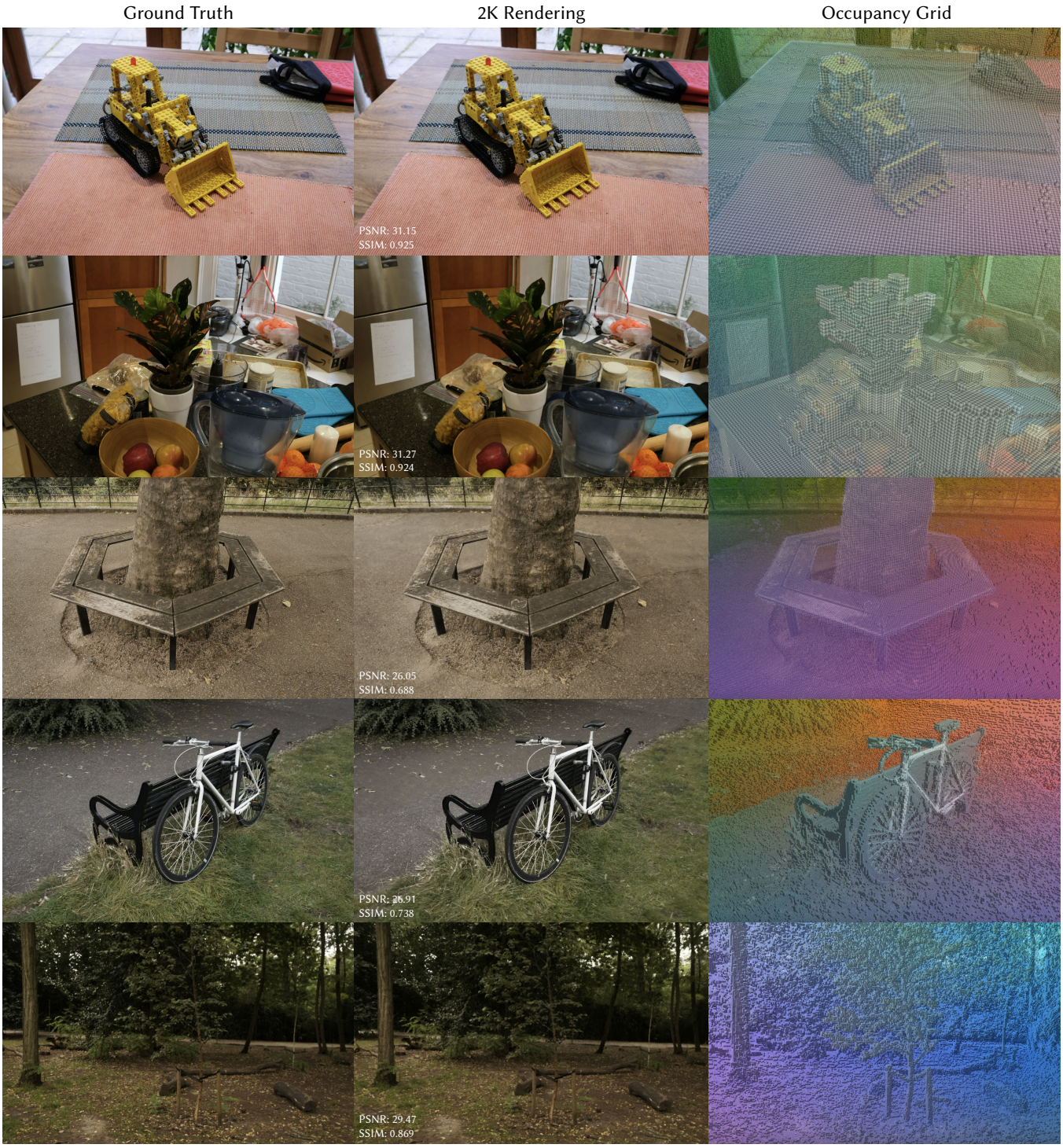


Figure 25: Additional results on Mip-NeRF 360 scenes [Barron et al. 2022], trained on 2K resolution images for 50K iterations. (Best zoom in to investigate details.)

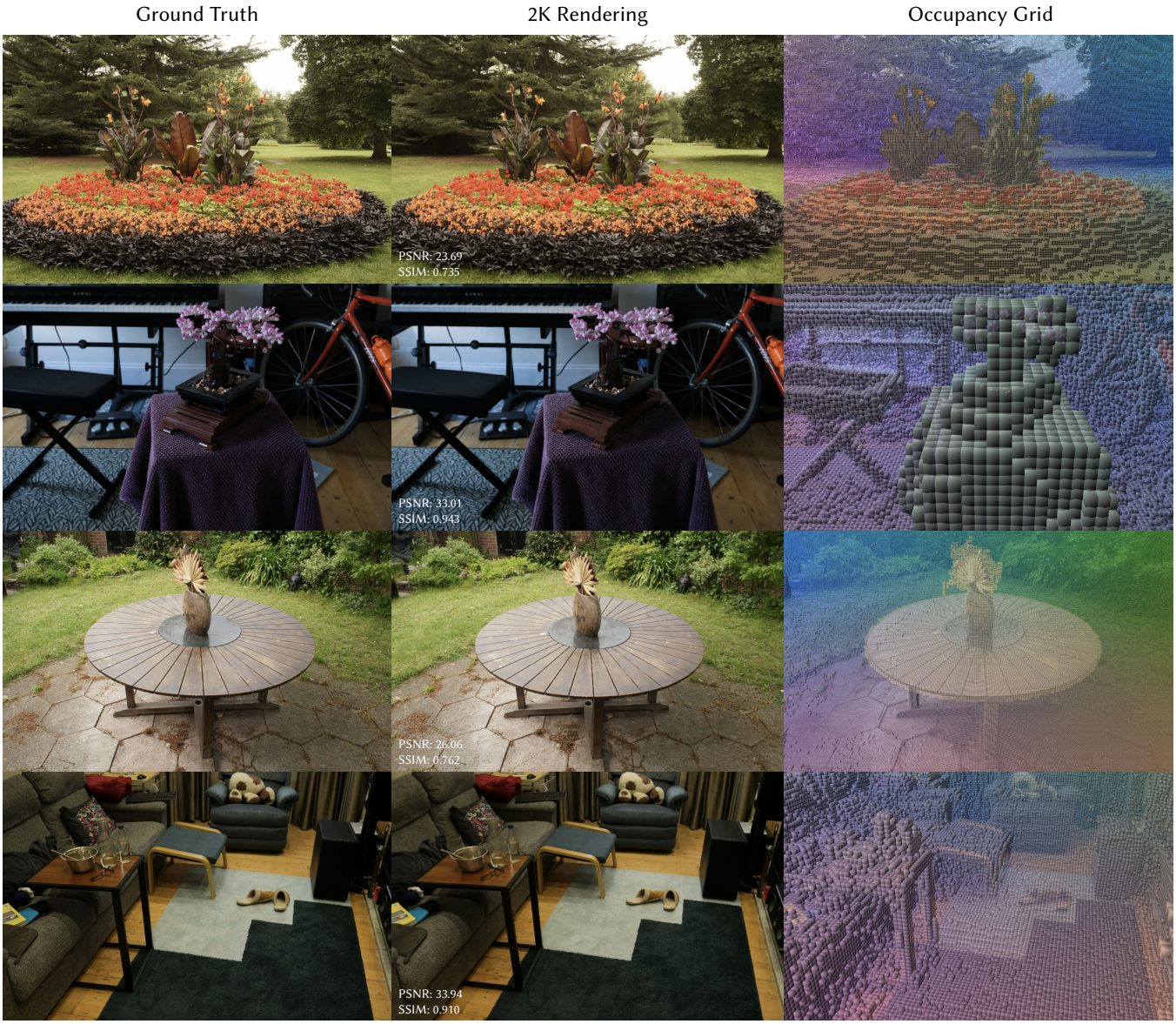


Figure 26: Additional results on Mip-NeRF 360 scenes [Barron et al. 2022], trained on 2K resolution images for 50K iterations. (Best zoom in to investigate details)

C.5 High-resolution rendering

Figure 23 shows the learning curve of 1K and 2K training results. The validation PSNR for each 1K & 2K pair is generally close, with slight drop in SSIM metric. Figure 9 shows an example trained with 4K resolution with fine-grained details.

C.6 Handling per-image variations

To explain per-image appearance variations, a latent code is commonly attached to each training image following the practice of Martin-Brualla et al. [2021]. One specialty of our captured data is that instead of using per-image latent code, we can consider using per-frame latent code (shared by 22 cameras at a same time) as a stronger regularization constraint. During our capture process, the outdoor lighting conditions can change slightly, and the moving people and capture rig can cast annoying shadows sometimes. It still remains an open question for us how to deal with these shadows effectively, as we found that using the interpolated latent code (Figure 28) or modeling with shadow field [Wu et al. 2022b] explicitly (Figure 29) can only lead to sub-optimal solutions. This becomes an extreme challenging scenario when we only have few images for each observation locations while most of them are cast by shadows.

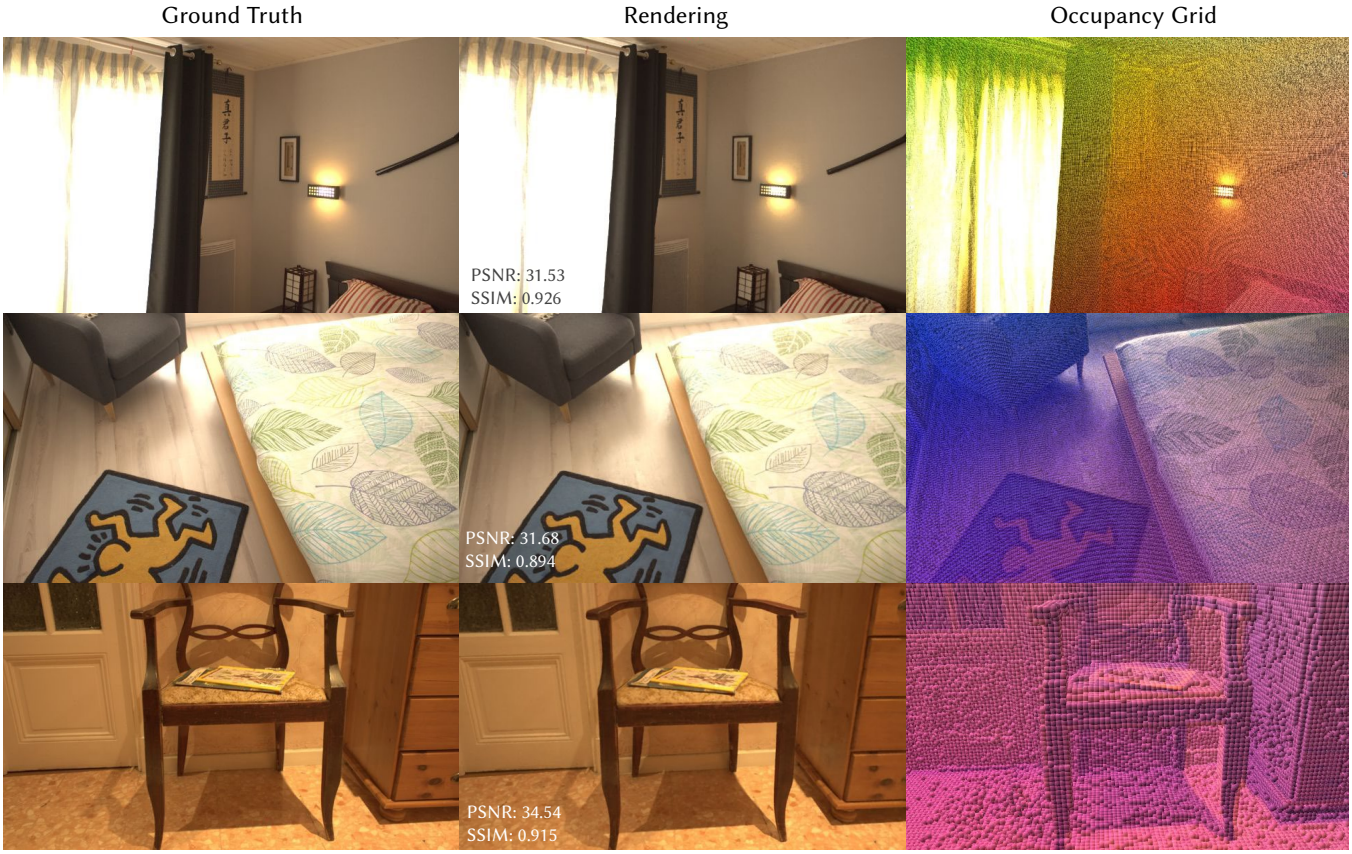


Figure 27: Additional results on Inria scenes [Philip et al. 2021], trained on 1K resolution images for 100K iterations. (Best zoom in to investigate details)

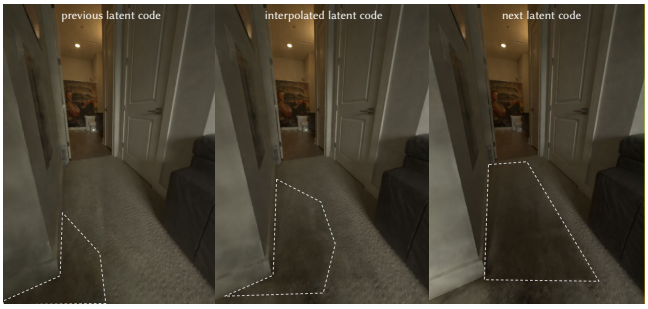


Figure 28: Latent code condition. We try to use latent code to explain away shadow issues. The image is rendered at the interpolate frame between two adjacent capture timestamps. From left to right we show results of using the learned latent code from previous frame, interpolated latent code, and latent code from next frame. While showing the tendency of moving the shadows smoothly (highlighted in white polygons), the overall appearance remains noisy with dark floats.

Figure 29: Shadow Fields. We implemented a shadow field [Wu et al. 2022b] to explain shadows with per-point predicted attenuation. We found it helpful to use low-frequency grid feature only for shadow field prediction. While the predicted shadow field looks reasonable in general, the accuracy is not sufficient to properly compensate for the affected appearance.

C.7 Nerfstudio results (nerfacto)

Pure MLP-based NeRF methods have difficulty in scaling up due to slow training and limited model capacity. An alternative baseline we considered is the versatile nerfstudio tool [Tancik et al. 2023]. We test a subset of our dataset and use nerfstudio for training and evaluations. We use the integrated nerfacto model, trained on 2K image for 100K iterations. The parameter setting for hash grid matched with our model ($128 \cdot 1.4^{15}$), and leave other components

Table 4: Test results on selected datasets using nerfstudio [Tancik et al. 2023]. PSNR/SSIM/LPIPS and FPS are reported here.

	PSNR	SSIM	LPIPS	FPS
SEATING_AREA	30.01	0.891	0.140	0.517
WORKSHOP	26.85	0.849	0.277	0.570
OFFICE2	26.60	0.939	0.127	0.595
OFFICE_VIEW1	27.01	0.804	0.382	0.374
RIVERVIEW	26.66	0.817	0.260	0.345

Figure 30: Qualitative comparison between ours and nerfacto. While the nerfacto model tends to produce scenes with smooth geometry and visuals, our model renders finer details in terms of correct color and high-frequency details, as visible in the zoomed-in patches.

with default configurations. The poses are processed with the XML camera pose file produced by Agisoft Metashape. We use the fisheye lens model for Eyeful Tower v1 images (with cropped black borders), and perspective model for Eyeful Tower v2 images, which are supported directly by nerfstudio.

Table 4 and Figure 30 show the quantitative and qualitative results for nerfstudio nerfactor model trained with sRGB spaces. We found that they can handle far-field well in unbounded scenes and can capture most details in the scene, yet commonly miss the detailed textures such as those on the carpets and floor. Note that our results shown in Figure 30 are trained on HDR and converted to sRGB space, where we can fairly compare the rendered visual quality.

D ADDITIONAL RENDERER DETAILS

D.1 Design of our 20-GPU Workstation Machine

The design of our custom 20-GPU rendering workstation was driven by the following goals:

- **Single CPU.** Our early experiments revealed higher stability for the Oculus VR runtime with single-socket computers than with dual-socket machines. Additionally, programming for dual-socket machines requires special considerations, e.g. when crossing NUMA domains. To maximize stability and minimize programming difficulty, we require a single CPU.
- **16+ direct-attached dual-slot GPUs.** The GPUs should be available to programs just as the typical 2–4 are on workstations, i.e., without network access or special cluster management software, as a typical render farm would have. The GPUs should be approximately equivalent to desktop Nvidia RTX 3090 cards. This, combined with the previous goal, should enable applications written for our multi-GPU workstations to fully utilize the machine with no code changes.
- **Windows 10 OS.** The Oculus VR stack only works on Windows operating systems. Using Windows 10 (instead of e.g. Windows 11 or Windows Server) allows the machine to more closely match our development workstations.
- **Mobile and quiet.** The machine should live inside a movable enclosure that can fit through a standard 32" door, and be quieter than 55 dB within it. This enables the machine to be taken to conferences and for demos to be given in the same room.

A thorough survey of commercial options found no solutions which satisfy all above requirements. Many vendors offer workstations or servers with 8 GPUs, but nearly all use two CPU sockets. A few vendors offer 10–16 GPU servers, but these are typically limited to single-slot GPUs, and always use two sockets. Thus, we build our own solution. This system is housed in a USystems Edge 3 sound-dampening rack that offers 30 dB of noise reduction.

D.2 Efficient Level-of-Detail Rendering

Section 4.2 described the advantages that level-of-detail rendering can have on image quality. However, our LOD-based masking strategy can also improve rendering performance. Kernel profiling measurements revealed that substantial time per frame is being spent waiting for features to be sampled from the multi-resolution hash grid, with the largest portion of the time being taken by the finest feature layers. This is likely due to the hash grid storage that leads to highly incoherent memory accesses. Coarser levels are stored within a dense linear array, and do not suffer as much from this issue, though the memory layout is still suboptimal for spatial coherence. Our LOD-based masking strategy removes the need to sample many of these expensive hash-grid features, and thus decrease rendering time, as we're able to conditionally replace the finest sampled levels with zeros.