

# NeLF-Pro: Neural Light Field Probes for Multi-Scale Novel View Synthesis

Zinuo You<sup>1</sup>    Andreas Geiger<sup>2</sup>    Anpei Chen<sup>†,1,2</sup>  
<sup>1</sup>ETH Zürich    <sup>2</sup>University of Tübingen, Tübingen AI Center  
 zinyou@ethz.ch    {a.geiger, anpei.chen}@uni-tuebingen.de

## Abstract

We present *NeLF-Pro*, a novel representation to model and reconstruct light fields in diverse natural scenes that vary in extent and spatial granularity. In contrast to previous fast reconstruction methods that represent the 3D scene globally, we model the light field of a scene as a set of local light field feature probes, parameterized with position and multi-channel 2D feature maps. Our central idea is to bake the scene’s light field into spatially varying learnable representations and to query point features by weighted blending of probes close to the camera - allowing for mipmap representation and rendering. We introduce a novel vector-matrix-matrix (VMM) factorization technique that effectively represents the light field feature probes as products of core factors (i.e., VM) shared among local feature probes, and a basis factor (i.e., M) - efficiently encoding internal relationships and patterns within the scene. Experimentally, we demonstrate that *NeLF-Pro* significantly boosts the performance of feature grid-based representations, and achieves fast reconstruction with better rendering quality while maintaining compact modeling. Project page: <https://sinoyou.github.io/nelf-pro>.

## 1. Introduction

Efficiently representing and reconstructing 3D scenes that support high-fidelity novel view synthesis is critical for computer vision and graphics applications, including virtual and augmented reality, cinematography, robotics and autonomous navigation. Neural Radiance Fields (NeRF) [38] have emerged as an effective representation to model real-world objects/scenes by incorporating density with view-dependent radiance.

Despite the high rendering quality, NeRF takes days to reconstruct an object or scene. The follow-up works demonstrated significant success in speeding up the training process by relaxing the input domain to optimizable feature grids [2, 12, 39, 48]. To fit an unbounded

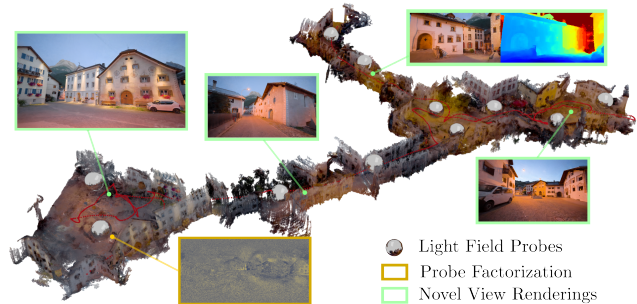


Figure 1. **NeLF-Pro** represent a **Scene** as spatially distributed **Light Field Probes** for faithful novel view synthesis in diverse and spatially inhomogeneous natural scenes.

scene into bounded MLP and feature grids, scene contraction [6, 39, 58, 68] is commonly used to warp the input domain within a ball or cube of fixed size. Despite their impressive performance, the basic representations they use, such as MLPs [6], VMGrids [12], and Hash-Grids [39, 58], are constrained to small-scale scenes (e.g., mip-NeRF360 and Free dataset), because these representations rely on a bounded centralized space. To represent large-scale scenes, a common strategy is to subdivide the target scene into cells, each endowed with individual representations [18, 35, 49, 54, 62]. However, determining the ideal position and dimensions of the local bounding boxes is not trivial. Employing bounding boxes with hard partitions can precipitate unstable optimization, giving rise to “floater” artifacts as demonstrated in KiloNeRF [44]. To solve this, regularization terms such as geometrical pixel partition [54] or total variance loss [63] are usually needed to stabilize the reconstruction. On the other hand, soft partitions with overlaps require smoothing filters, such as appearance matching [49] or per-frame blending [35] for the colors predicted from different local representations, resulting in slow optimization.

In this work, we propose **Neural Light Field Probes** (aka, **NeLF-Pro**), a novel representation that is compatible with small and large scale scenes, and at the same time achieves on-par-with or better-than current state-of-the-art novel view rendering quality. Unlike previous methods that

† Corresponding author.

encode positions globally into multilevel inputs, we decompose the scene into local probes, allowing for scale- and trajectory-agnostic modeling. The core idea of our representation is to jointly encode the light field and geometry using Light Field Probes and optimize the probes by gradient descent. Unlike standard light field methods that model rays using global parametrization, our light field probes capture visible rays in local areas, achieving occlusion-aware modeling and rendering.

To do so, we first decompose the global coordinates into local ones. We observe that encoding a large scene globally within a compact representation, such as VMGrid or Hash-Grid, necessitates a substantial number of factors or levels. Consequently, the entire representation must be loaded when querying features, leading to inefficiencies in computations and GPU memory utilization. To address these issues, we propose to decompose the scene into multiple local representations based on the camera trajectory. This leads to flexible scene modeling when dealing with diverse natural scenes that vary in extent and spatial granularity.

We leverage the fact that a full light field can be seen as an integration of light field samples, where each light field sample can be approximated by blending a collection of discrete light field images - which have been widely applied in Image Based Rendering (IBR) [27]. We therefore present a novel light field feature probe representation to encode light field radiance and geometry as multiple-channel spherical feature maps. In this case, light field samples can be approximated by aggregating probe features. Unlike the recent popular scene contraction mechanism [6], our work relaxes the assumption of a single scene center by retaining multiple representation centers. With this approach, we locally encode the radiance and geometry of the scene, thereby facilitating decomposition and reconstruction.

Our work aims to improve grid-based representations to handle scenes with arbitrary scales within a compact and efficient framework. Towards this goal, we present a novel **Vector-Matrix-Matrix** (VMM) decomposition technique that effectively represents the light field feature probes as the products of a core factor (i.e., VM) which is shared among local feature probes, and a basis factor (i.e., M) which is designated exclusively for each probe. The spatial sharing mechanism facilitates the uncovering of hidden relationships and patterns within the scene, while reducing the number of light field feature probes required to maintain expressivity. With VMM factorization, our work compactly encodes the entire scene content within dispersed light field feature probes across the spatial domain, achieving compact encoding while also preserving locality.

We further propose a soft localization and blending algorithm to enable fast reconstruction and aliasing-free rendering. Specifically, we aim to construct a target image by selecting  $K$  neighboring feature probes based on the dis-

tance between the target camera center and the probe centers. Then, instead of representing each probe feature individually through factor multiplication, we apply efficient weighted blending on each factor group before multiplication. It should be noted that our feature query differs in a principled way from previous grid-based representations. We index point features on camera-adjacent probes rather than point-adjacent probes. Together with the local spherical projections, this enables storing of a 3D point at different probes in a form of multi-scale pyramid (also known as mipmaps) - nearby areas are modeled at higher resolution, while distant areas are represented at lower resolution. This conforms to the perspective projection and allows querying the mip feature without tracing cones [6, 20] or multiple samples [7]. Additionally, the neighboring probe selection scheme demonstrates 3D scene locality and is able to facilitate training and rendering on a large scale without the need to load the entire representation into memory. For continuous light field factorization, as inspired by TensorRF [12], we apply linear factor interpolation for the query points.

We conduct extensive experiments on diverse scenes, including both small-scale, i.e., mip-NeRF360 [6], Free [58], middle-scale KITTI360 [29] dataset) and large-scale, i.e., Google Earth [62], KITTI-360-large [29] and website video. We demonstrate that all models are able to achieve realistic novel view synthesis results that are on-par-with or better than state-of-the-art methods, especially on large-scale scenarios, providing noticeable improvements while preserving fast reconstruction and high compactness.

## 2. Related Works

**Light fields.** Light fields capture the spatial-angular distribution of radiance in a scene. They were first introduced by Levoy and Hanrahan and enable novel view synthesis without the need for explicit 3D models [27]. Gortler et al. [10, 16, 19, 61] extended this idea with proxy geometry and improved the rendering quality using approximate geometry. The advent of neural networks brought a paradigm shift in light field rendering and early works [11, 41] use MLPs to implicitly model light fields and directly predict ray colors. [28, 56] propose novel light field representations for efficient rendering. Similarly, [37, 71] propose to predict blending weights based on image features and ray distances.

Directly representing a full light field with a neural network is extremely difficult because of its complexity. To address this problem, Light Field Networks [46] leverage an implicit MLP together with 6D Plücker parameterization of the space of light rays, allowing 360-degree light field modeling and demonstrating success in representing scenes and objects with simple texture. Subsequent works [4, 22, 47, 59, 60] incorporate light field rendering

with volumetric integration to achieve high fidelity rendering, especially in high reflectent and translucent regions. However, they are limited to forward-facing or small-scale synthetic scenes.

Light Field Probes were designed to encode a scene’s full light field and internal visibility by placing a probe array in the scene, allowing a quick rendering of global illumination effects [33, 45]. Our work is inspired by probe representation. However, unlike traditional rendering setups, our work assumes that the geometry and radiance of the scene are unknown and attempts to learn and reconstruct the scene with gradient descent. We refer the readers to [43, 51] for a more comprehensive view.

**Neural radiance field.** Advanced neural representations are revolutionizing the reconstruction field as a promising replacement for traditional representations. They show significant progress in improving reconstruction quality and efficiency in a wide range of graphics and vision applications, including novel view synthesis [3, 12, 30, 31, 38, 52, 55, 64, 71], surface reconstruction [40, 57, 65–67], appearance acquisition [8, 9, 21, 69, 70], etc. In contrast to standard explicit representations such as meshes, point clouds, implicit representations [15, 34, 42] provide continuous and smooth representations of surfaces.

NeRF [38] represents the neural radiance field for novel view synthesis using MLPs with positional encoding for scene representation. However, its reconstruction process is time-consuming. To address this, recent works have introduced learnable feature grids [2, 12, 39, 48] for faster optimization, achieving reconstructions in just 5 minutes. However, these representations, along with state-of-the-art methods [6, 7], assume object-centric scenarios and use scene contraction, which limits their scalability. To achieve compact reconstruction and constrain the reconstruction process, TensorRF uses various matrix factorizations to model a radiance field. However, it is scene-centric and fails to model large-scale scenes. In contrast, F<sup>2</sup>-NeRF [58] identifies a 3D subspace where the distance between any two points is equal to the sum of their 2D perspective distances across all visible cameras. However, F<sup>2</sup>-NeRF still depends on the HashGrid representation, which limits its suitability for long trajectories or large-scale scenes. Our approach, on the other hand, defines the target space as a set of local 2D subspaces (light field probes) rather than a homogeneous 3D space. This naturally preserves the perspective distances and provides flexibility when modeling scenes with different scales.

**Large-scale 3D reconstruction.** Numerous works have succeeded in reconstructing large-scale scenes using standard structure-from-motion techniques or recent neural radiance fields. Early work [1] presents a system that can match and reconstruct entire cities from images collected

from the Internet. While [26] improve accuracy, they still mainly focus on reconstructing point clouds with diffuse color, and use Poisson reconstruction to obtain meshes, lacking the ability to render realistic images from reconstructed representations.

Recent studies [35, 36, 49, 54, 62, 63] have integrated NeRF with the concept of decomposing large-scale scenes into a union of local blocks while maintaining individual representations for each block. In practice, this localized approach introduces optimization challenges, often requiring the incorporation of smoothing regularization or additional supervision to enhance the stability. To address challenges such as inconsistent exposure and moving objects in the capture of large scenes, NeRF-W [32] and Block-NeRF [49] present an uncertainty and latent appearance modeling, which significantly reduce the reconstruction “floaters”. It is worth mentioning that 3D Gaussian Splatting [23] doesn’t rely on scene contraction but with a projection mechanism, showing potential for large-scale scene reconstruction. However, the primitive-based representation can easily overfit to the training views and thus requires a good initial point cloud and dense training views. Our representation is between local radiance fields and primitive representations, utilizes only a few dozen local probes, and does not require a point cloud prior.

### 3. Neural Light Field Probes

Our goal is to synthesize the novel views of the diverse scenes that vary in scale and spatial granularity, from a collection of posed multi-view images. In the following sections, we first present our scene parameterization which bridges the gap between light field probes and volume rendering. We then introduce a factorization algorithm that efficiently factorizes radiance and geometric information for light field queries. Finally, we describe in detail how to distribute the probes during initialization and select them during querying for high-fidelity rendering.

#### 3.1. Volumetric Light Field Rendering

A continuous light field  $\mathcal{L}(\mathbf{x}, \mathbf{d})$  encodes the spatial-angular distribution of radiance for all receivers  $\mathbf{x} \in \mathbb{R}^3$  over the unit outgoing directions  $\mathbf{d} \in \mathbb{S}^2$ . We build a sparse set of discrete representations of this distribution:  $\mathcal{L}_l(\mathbf{d}) := \mathcal{L}(\mathbf{p}_l, \mathbf{d}), l \in [1, \dots, L]$  (*Light Field Probes*), where each probe captures the angular variation as seen from the probe position  $\mathbf{p}_l$ . The light field at any receiver  $\mathbf{x}$  can be reconstructed by calculating a spatial average from the radiance of the probes.

$$\mathcal{L}(\mathbf{x}, \mathbf{d}) \approx \sum_{l=1}^L w_l(\mathbf{d}) \mathcal{L}_l(\mathbf{d}) \quad (1)$$

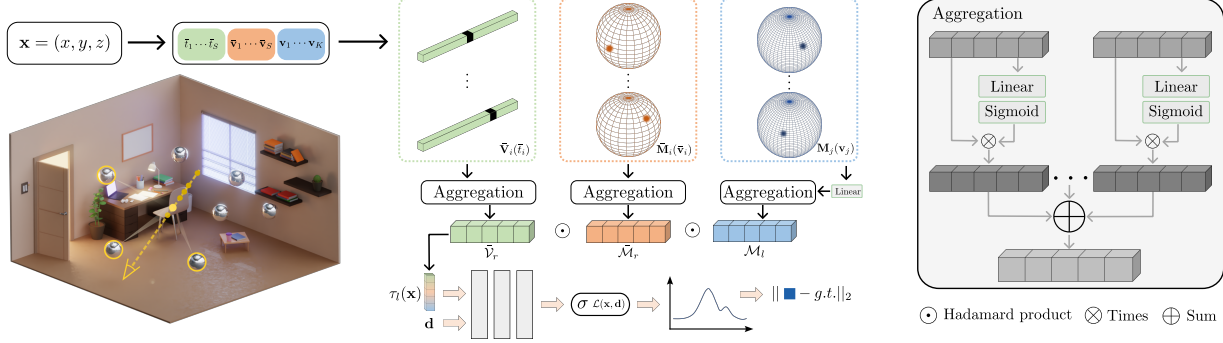


Figure 2. **NeLF-Pro pipeline.** Our approach models a scene using light field probes represented by a set of core vectors  $\bar{\mathbf{V}}$ , matrices  $\bar{\mathbf{M}}$  as well as a set of basis matrices  $\mathbf{M}$ . We project the sample points onto local probes to obtain local spherical coordinates and then query factors from probes near the camera. These factors are aggregated using a self-attention mechanism and combined using Hadamard products. The resulting factor  $\tau_l(\mathbf{x})$  and the viewing direction  $\mathbf{d}$  are used to calculate the density  $\sigma$  and the light field radiance  $\mathcal{L}(\mathbf{x}, \mathbf{d})$ .

where  $w_l(\mathbf{d}) = \tilde{w}_l(\mathbf{d}) / \sum_k \tilde{w}_k(\mathbf{d})$ , which denotes the normalization weighting depending on the distance between  $\mathbf{x}$  and  $\mathbf{p}_l$ .

Note that the main difference between light field and radiance field [38] rendering is that, light field rendering directly utilizes the distribution of light in ray space, while radiance field reconstructs the light distribution in a volumetric space by integration.

Volumetric light field rendering combines the advantages of both methods and proposes to divide the target ray into a sequence of intervals between the near and far ranges and to sum over the light field segments. The light segment of each interval is determined by weighted blending of light field samples from the multi-view images:

$$\mathcal{L}(\mathbf{x}, \mathbf{d}) \approx \sum_{i=1}^N T_i \alpha_i \mathcal{L}(\mathbf{x}_i, \mathbf{d}) \quad (2)$$

$$\text{where } \mathcal{L}(\mathbf{x}_i, \mathbf{d}) = \sum_{l=1}^L w_{i,l} \mathcal{L}_l(\mathbf{v}_{i,l}) \quad (3)$$

where  $T_i$  are the approximated accumulated transmittances defined by  $T_i = \prod_{j=1}^{i-1} (1 - \alpha_j)$ , and  $\alpha_i = 1 - \exp(-\sigma_i \Delta_i)$  is the approximated opacity value of sub-interval  $\Delta_i = t_{i+1} - t_i$ , and  $\sigma_i$  denotes the volume density of point sample  $\mathbf{x}_i = \mathbf{x} + t_i \mathbf{d}$ .  $\mathbf{v}_{i,l} = (\mathbf{x}_i - \mathbf{p}_l) / \|\mathbf{x}_i - \mathbf{p}_l\|_2$  is the normalized direction between the sample point and the position of light field probes,  $w_{i,l}$  are the normalized blending weight of the intervals.

To obtain the density and blending weight at position  $\mathbf{x}_i$ , a common solution is to reuse the input images as light field samples and to train a neural network  $\Phi$  to predict the density and blended radiance from a set of light field queries  $\mathcal{L}_l(\mathbf{x}_i, \mathbf{v}_{i,l})$ :

$$\alpha_i, w_{i,l} = \Phi(\{\mathcal{L}_l(\mathbf{v}_{i,l})\}) \quad (4)$$

However, these neural Image Based Rendering methods like IBNet [59] can easily lead to jumping artifacts when moving the camera and fail to accurately estimate geometry due to appearance bias. Instead of directly using images, we introduce light probes to model light field samples. Our work assumes that the fusion network  $\Phi$  and the light field probes  $\mathcal{L}_l(\mathbf{v})$  are both unknown and optimizes these parameters by gradient descent. In practice, the light field probes and the fusion network can easily be overfitted to the multi-view inputs due to the strong flexibility and view-dependent nature of the light field.

### 3.2. Light Field Factorization

We now present our novel light field probe representation for stable light field reconstruction. Our model consists of spatially distributed feature probes, each representing the radiance and geometry as multi-channel features.

We consider the discrete spherical probes as a set of 3D tensors  $\mathcal{T} = \{\tau_l \in \mathbb{R}^{H \times W \times D}\}_{l=1}^L$ , where each local probe  $\tau_l$  denotes a  $D$ -layered spherical grid located at  $\mathbf{p}_l$ ,  $W, H$  are the spherical dimensions, providing full angular coverage. We propose to use low-dimensional factors to represent the light field tensors  $\mathcal{T}$  for compact modeling, and introduce a novel vector-matrix-matrix (VMM) factorization, where a factor  $\bar{\mathbf{V}}$ ,  $\bar{\mathbf{M}}$ , serves as core factors and captures the multilinear interactions between different factors, accompanied by basis factors  $\mathbf{M}$  located at position  $\mathbf{x}$ . VMM factorization factorizes each tensor element  $\tau_l^{ijk}$  as a sum of  $R$  scalar products:

$$\tau_l^{ijk} = \sum_{r=1}^R \bar{\mathbf{V}}_{c,r}^k \circ \bar{\mathbf{M}}_{c,r}^{ij} \circ \mathbf{M}_{l,r}^{ij} \quad (5)$$

where the factor  $\bar{\mathbf{V}}_{c,r} \in \mathbb{R}^D$  is a vector of length  $D$ , and the matrix  $\bar{\mathbf{M}}_{c,r} \in \mathbb{R}^{H \times W}$ ,  $\mathbf{M}_{l,r} \in \mathbb{R}^{H \times W}$  are matrix tensors,  $\circ$  denotes the Hadamard product. We use  $c, l$  as the index

indices core and basis factors respectively,  $i, j, k$  denote the indices of the three modes.

**Distributing factors** We distribute both the core and basis factors as scene representation in 3D, and apply the Farthest Point Sampling (FPS) [17] algorithm to select  $L$  positions for the basis factor from the input cameras, and then use the K-means algorithm to select  $C$  positions for the core factors. The number of factors  $C, L$  can be set differently and should be chosen depending on the complexity of the scene. Inspired by the factor sharing mechanism in Tucker Decomposition [53], we propose to share the core factor across different basis factors, resulting in  $C \ll L$ . The sharing mechanism enables the discovery of hidden relationships and patterns within the scene.

Unlike previous scene representations, which are usually centered at the scene, our approach exploits the prior information provided by the input camera positions, facilitating the handling of scenes with arbitrary shapes. By distributing local scene representations in these regions, our method enhances representation efficiency.

**Continuous factorization.** To achieve a continuous field, we employ linear interpolation on each vector and matrix while querying the tensor grid features of a target position  $\mathbf{x}$ , combining with Eq. (5) we obtain,

$$\tau_l(\mathbf{x}) = \sum_{r=1}^R \bar{\mathbf{V}}_{c,r}(\mathbf{x}) \circ \bar{\mathbf{M}}_{c,r}(\mathbf{x}) \circ \mathbf{M}_{l,r}(\mathbf{x}) \quad (6)$$

where  $(\cdot)$  denotes linear interpolation.

**Coordinate transformation.** To query values on the low-dimensional factors, we transform the input coordinates  $\mathbf{x}$  to local coordinates using a projection transformation. For a 3D location denoted by  $\mathbf{x} = (x, y, z)$ , we compute its distance to a factor center  $\tilde{t} = \|\mathbf{x} - \mathbf{p}\|_2$  and transform it into local normalized depth and spherical coordinates as follows:

$$t = \frac{1}{\tilde{t} + 1}, \tilde{\mathbf{v}} = \left( \arccos\left(\frac{z}{\tilde{t}}\right), \arctan\left(\frac{y}{x}\right) \right) \quad (7)$$

Following Factor Fields [13, 14], we apply a scaling factor and a periodic function to the projected coordinate to effectively utilize the empty regions within the spherical grids,  $\mathbf{v} = \gamma(\nu \cdot \tilde{\mathbf{v}})$ , where  $\nu$  determines the frequency and  $\gamma$  is the sawtooth function.

Therefore, the tensor values at location  $\mathbf{x}$  denoted in Eq. (6) are given by:

$$\tau_l(\mathbf{x}) = \sum_{r=1}^R \bar{\mathbf{V}}_{c,r}(\tilde{t}_c) \circ \bar{\mathbf{M}}_{c,r}(\tilde{\mathbf{v}}_c) \circ \mathbf{M}_{l,r}(\mathbf{v}_l) \quad (8)$$

where  $\tilde{t}$  and  $\tilde{\mathbf{v}}$  represent the projected coordinates of the core factors.

To simplify Eq. (8), we concatenate all scalar values into an  $R$ -channel vector, this yields  $\bar{\mathbf{V}}_c(\tilde{t}_c) = [\bar{\mathbf{V}}_{c,1}(\tilde{t}_c), \dots, \bar{\mathbf{V}}_{c,R}(\tilde{t}_c)]$ . Similarly, we define  $\bar{\mathbf{M}}_c(\tilde{\mathbf{v}}_c)$  and  $\mathbf{M}_l(\mathbf{v}_l)$ . In the following, we denote  $\mathcal{G}_l(\mathbf{x}) = \bar{\mathbf{V}}_c(\tilde{t}_c) \circ \bar{\mathbf{M}}_c(\tilde{\mathbf{v}}_c) \circ \mathbf{M}_l(\mathbf{v}_l)$ . If necessary, we can recover  $\tau_l(\mathbf{x})$  from  $\mathcal{G}_l(\mathbf{x})$  using a simple sum operation  $\tau_l(\mathbf{x}) = \sum \mathcal{G}_l(\mathbf{x})$ , though this operation is omitted with the inclusion of a neural decoder.

**Localization.** We now detail how to form an image using a subset of probes, resulting in low memory consumption and scalable representation. When the scene size and consequently the number of necessary probes increases, rendering becomes inefficient.

Therefore, we propose to select the  $\hat{C}$  nearest core factors and the  $\hat{L}$  basis factors out of  $C$  and  $L$  based on the distance between  $\mathbf{p}_l$  and the position of the target camera. After that, only the selected factors are used when rendering a target view, enabling reconstruction and rendering of large scenes or using computationally restricted hardware.

In addition, such a strategy implicitly mimics a mipmap representation. Due to the spatially distributed light field probes and spherical projection transformation, the light field features of a 3D point are captured at multiple resolutions across different probes, where the representation resolution is automatically adjusted based on the relative distance between the 3D point and the probe's center. Rendering with the feature in neighboring probes naturally encodes a proper level of scene features, where 3D content close to the target view is represented at high resolution due to the nature of the spherical projection, as described in Section 3.2.

**Factor aggregation.** Our work leverages local modeling and regionally selects factors for target view rendering. Naively concatenating the selected factors can introduce a permutation-variant problem. To address this problem, we propose a permutation-invariant factor blending scheme. Given the  $\hat{C}$  selected shared core and  $\hat{L}$  basis factors for each query point, we predict a weight for each factor and blend the factors of same groups using a self-attention mechanism:

$$\mathcal{G}(\mathbf{x}) = \bar{\mathcal{V}} \circ \bar{\mathcal{M}} \circ \mathcal{M}, \text{ where } \begin{cases} \bar{\mathcal{V}} = \sum_{i=1}^{\hat{C}} w_i \bar{\mathbf{V}}_i(\tilde{t}_i) \\ \bar{\mathcal{M}} = \sum_{i=1}^{\hat{C}} w_i^{\bar{\mathcal{M}}} \bar{\mathbf{M}}_i(\tilde{\mathbf{v}}_i) \\ \mathcal{M} = \sum_{j=1}^{\hat{L}} w_j^{\mathcal{M}} \mathbf{M}_j(\mathbf{v}_j) \end{cases} \quad (9)$$

where the  $\mathcal{G}(\mathbf{x}) \in \mathbb{R}^R$  is the aggregated vector.

we obtain the scalar blending weight  $w$  by a trainable linear transformation with sigmoid activation,

$$w = \text{Sigmoid}(\mathbf{W}f + b) \quad (10)$$

here,  $f = \bar{\mathbf{V}}_i(\bar{t}_i)$ ,  $\bar{\mathbf{M}}_i(\bar{\mathbf{v}}_i)$ , and  $\mathbf{M}_j(\mathbf{v}_j)$  respectively.

To further boost the compactness, we reduce the component dimension of the basis factor from  $R$  to  $F$  by applying a linear layer to  $\mathbf{M}_j(\mathbf{v}_j)$  before factor aggregation, as shown in Fig. 2.

**Decoder.** So far, we have introduced the factorization of the light field features. However, unlike the direct light field rendering in Eqn.(2), which works with radiance samples, our light field representation is embedded in feature space. To obtain multi-dimensional outputs, we introduce a MLP network  $\mathcal{P}$  to obtain the radiance and geometric density:

$$\alpha_i, \mathcal{L}(\mathbf{x}_i, \mathbf{d}) = \mathcal{P}(\mathcal{G}(\mathbf{x}_i), \mathbf{d}) \quad (11)$$

### 3.3. Optimization

We optimize the light field feature probes and the projection network using gradient descent, and minimize a  $L_2$  loss between the reconstructed images and the multiview inputs. Using global and local factorization as basis factors that correlate and regularize the entire field during optimization, our method achieves efficient reconstruction. Furthermore, we include the interlevel loss from mip-NeRF 360 [6] in our optimization process.

## 4. Experiments

### 4.1. Experimental Settings

**Datasets** We evaluate our method on four datasets spanning diverse scene scales and camera trajectories, including the mip-NeRF360 dataset [6], the Free dataset [58], BungeeNeRF [62], and KITTI-360 [29], as well as a city walk video from [Youtube](#), their trajectories are shown in Fig. 3. For the KITTI-360 dataset, we use two different scales: medium and large scales. For medium-scale scenes, we used five static sequences from Novel View Appearance Synthesis (50% Drop), with a distance of about 50 meters, labeled as KITTI-360. In addition, we selected a long trajectory with a driving distance of approximately 1500 meters for a large-scale evaluation, labeled as KITTI-360-big.

**Implementations** We implement our NeLF-Pro with PyTorch within the SDFStudio framework [50, 66], without customized CUDA kernels. We train our model on a single RTX 3090 GPU with Adam optimizer [24] and a learning annealed in multiple steps from  $10^{-2}$  to  $3.5 \times 10^{-4}$ . We use a two-layer proposal model [6] for sampling points along the rays, with [256, 96, 48] samples per stage. The proposal model is represented with our sparse core factor and is optimized by a distillation loss [6].

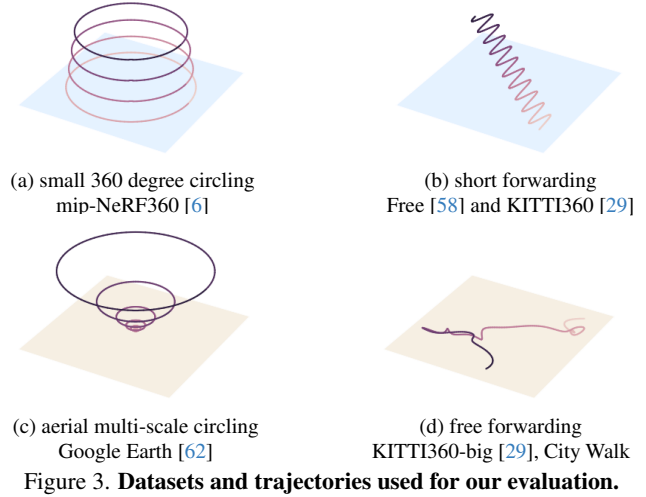


Figure 3. **Datasets and trajectories used for our evaluation.**

We implement the core and basis factors of light field probes with feature grids. We set the resolution of the core vector  $\bar{\mathbf{V}}$  to 64 and the matrix  $\bar{\mathbf{M}}$  to (64, 128), progressively upscaling them to 1024 and (128, 256), respectively. For the basis factor, we opt for a lower resolution, fixed at (256, 512) during the optimization process. For all experiments, we maintain the factor component number at  $R = 32$  and  $F = 2$ .

Due to the difference in scale, the probe number  $L$  and the core tensor number  $C$  vary for different datasets. For small scale scenes, we set  $(L, C) = (64, 3); (100, 5); (256, 16)$  for the small, medium, and large scale scenario respectively. We use  $\hat{L} = 16$  neighboring basis factors to represent the target views. The core factor for neighbor selection  $\hat{C}$  is 3 for centralized camera trajectories (mip-NeRF360 and 56Leonard) and 2 for others. The periodic factor  $\nu$  is (5, 5) for the mip-NeRF360 dataset, (5, 2) for KITTI middle scale and (4, 4) for others.

### 4.2. Comparisons

**Small Scale Scenes.** We consider the mip-NeRF360 and Free datasets as small-scale datasets and compare our method with MLP and primitive-based representations, partly with voxel-grid methods, as shown in Table 1.

Our NeLF-Pro significantly minimizes the known performance gap between feature grid-based representations and the well-trained MLP SOTA on the unbounded scenes, achieving SOTA performance over feature grid-based methods [39, 58] on both the Free and mip-NeRF360 datasets, while it is on-par-with mip-NeRF360 [6] on the Free dataset.

The qualitative results are shown in Fig. 4, our approach is capable of providing better renderings, especially in recovering thin structures. Thanks to the local representation, our method models specular reflections more accurately.

Method	Iterations	Batch Size	Free Dataset					mip-NeRF360 Dataset		
			Time ↓	#Params ↓	PSNR↑	SSIM↑	LPIPS ↓	PSNR↑	SSIM↑	LPIPS ↓
NeRF++ [68]	250K	2K	hours	-	23.47	0.603	0.499	26.21	0.729	0.348
mip-NeRF360 [6]	250K	16K	hours	8.6M	<b>27.01</b>	<b>0.766</b>	<b>0.295</b>	<b>28.94</b>	<b>0.837</b>	<b>0.208</b>
Plenoxels [2]	128K	5K	25m	Dynamic	19.13	0.507	0.543	23.35	0.651	0.471
DVGO [48]	30K	5K	21m	153M	23.90	0.651	0.455	25.42	0.695	0.429
Instant-NGP [39]	20K	10-85K	6m	12M	24.43	0.677	0.413	26.24	0.716	0.404
F <sup>2</sup> -NeRF [58]	20K	10-85K	12m	18M	26.32	0.779	0.276	26.39	0.746	0.361
NeLF-Pro (ours)	20K	4K	16m	20M	<b>27.02</b>	<b>0.802</b>	<b>0.256</b>	<b>27.27</b>	<b>0.753</b>	<b>0.360</b>

Table 1. **Results on the Free and mip-NeRF360 dataset.** The scores of the baseline methods are taken from F<sup>2</sup>-NeRF [58]. Our training times are evaluated on an RTX3090 GPU.

Method	PSNR↑	SSIM↑	LPIPS ↓	Time ↓
NeRF [38]	21.28	0.779	0.343	days
mip-NeRF [5]	21.54	0.778	0.365	days
PNF [25]	22.07	0.820	0.221	days
NeLF-Pro (ours)	<b>22.55</b>	<b>0.829</b>	<b>0.201</b>	45mins

Table 2. **Quantitative result on KITTI360-NVS-appearance-50% task [29].** We report SSIM and LPIPS from the leaderboard.

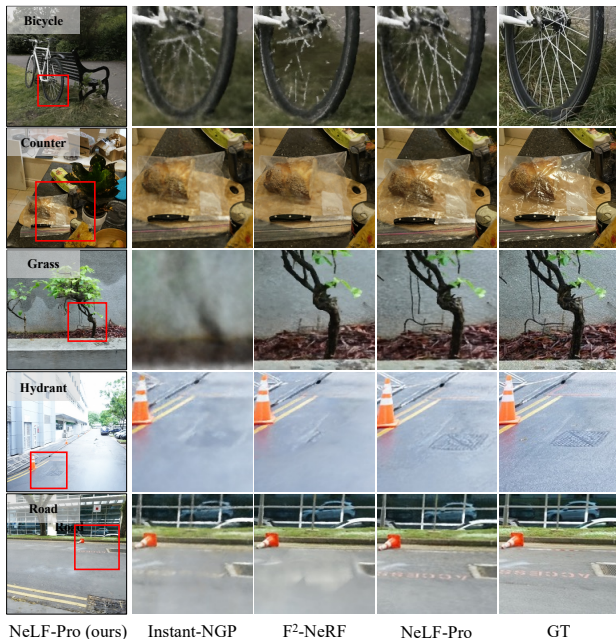


Figure 4. **Qualitative results on mip-NeRF360 [6] and Free dataset [58].** Our approach is able to better reconstruct thin structure and appearance details than the baseline methods (Instant-NGP [39], F<sup>2</sup>-NeRF [58]).

Furthermore, our method is efficient and shows a training time comparable to Instant-NGP [39] and F<sup>2</sup>-NeRF [58], both implemented with customized CUDA kernels. Note that F<sup>2</sup>-NeRF requires precomputing near-far values for each training view from the calibration point cloud, while we instead use constant near-far values without the need for a sparse point cloud.

**Medium Scale Scenes.** Since the training views are accessible for the KITTI 360 dataset, we submitted our re-

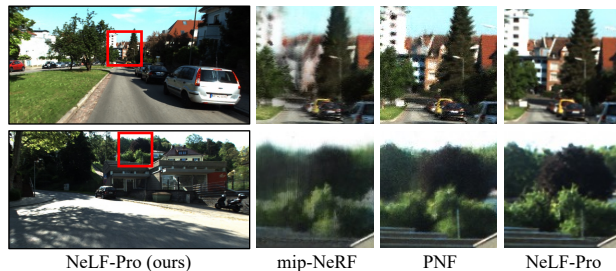


Figure 5. **Results on KITTI-360 NVS Task.** Our approach provides more sharp renderings than the baselines.

sults to the leaderboard to obtain evaluation scores. In Table 2, we compare our method with other submissions that used the same setting as ours, i.e., RGB supervision only. Note that, in compliance with KITTI-360’s guidelines, we couldn’t submit results for other methods. Thus, our comparison is limited to previous submissions. Our approach consistently outperforms all baseline methods for all three metrics and also provides more than 20 times faster training speed. Fig. 5 demonstrates that our approach yields sharper renderings compared to the baseline methods.

**Large Scale Scenes.** We also show the results of large-scale reconstruction for three distinctly different types of scenes: 56-Leonard with an input level of detail that includes a bird-eye-view, Scuol with a town in which randomly walking trajectory, and KITTI-360-big with a long straight driving style. In our experiments, we use a four-fold larger model with three-fold more iterations (60k iterations) to reconstruct large scenes. We applied a similar increase in model size to the comparison methods.

The quantitative Table 3 and visual results Fig. 6 clearly demonstrate the high efficiency and strong scalability of our approach. Impressively, we found that our approach is able to better preserve structured patterns, which benefit from local representation and modeling of core factors.

### 4.3. Ablation Study

In Table 4 we evaluate several ablations of NeLF-Pro: a-c) partial or complete removal of core factors  $\tilde{V}$  modeling; d) “point neighborhood” queries the nearest neighbor factors

Method	Time ↓	#Params ↓	56-Leonard			Scuol			KITTI-360-big		
			PSNR↑	SSIM↑	LPIPS ↓	PSNR↑	SSIM↑	LPIPS ↓	PSNR↑	SSIM↑	LPIPS ↓
mip-NeRF360 [6]	13.3h	8.6M	26.22	0.829	0.200	25.20	0.800	0.381	16.63	0.537	0.527
GridNeRF [63]	6.9h	276M	23.42	0.731	0.318	23.65	0.763	0.510	18.55	0.548	0.607
3DGS [23]	2.5h	225M	<b>29.23</b>	<b>0.942</b>	<b>0.075</b>	22.74	0.835	<b>0.301</b>	15.17	0.510	0.601
F <sup>2</sup> -NeRF-Big [58]	1h	148M	<u>27.07</u>	<u>0.930</u>	<u>0.102</u>	<u>26.43</u>	<u>0.842</u>	<u>0.385</u>	<u>21.36</u>	<u>0.686</u>	<u>0.446</u>
NeLF-Pro (ours)	1.6h	100M	<u>28.86</u>	<u>0.930</u>	0.108	<b>27.84</b>	<b>0.855</b>	<u>0.368</u>	<b>22.68</b>	<b>0.720</b>	<b>0.399</b>

Table 3. **We show averaged results for large-scale scenes.** **Bold** score for the first and underlined score for the second. Time and parameter number are the mean across all the scenes. In general, our method is on-par-with or outperforms other reconstruction methods. While 3DGS and F<sup>2</sup>-NeRF-Big achieve similar or better results than ours on the 56-Leonard scene, they perform better near the center of the scene and less effectively for zoomed-out views. In contrast, our approach maintains high rendering quality throughout the entire trajectory due to its distributed representation. Our representation is also more compact and requires fewer parameters.

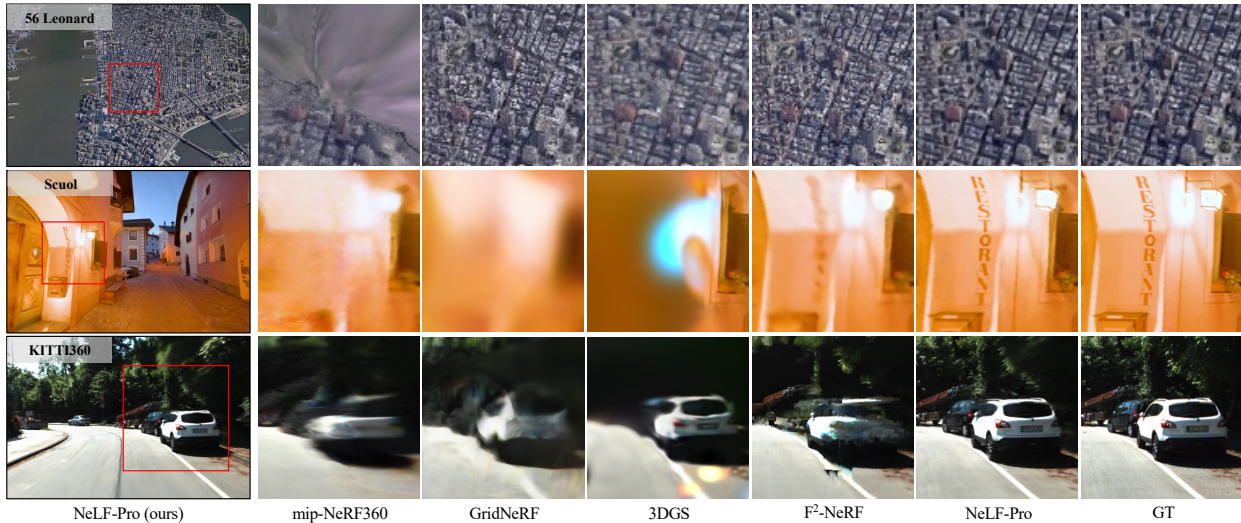


Figure 6. **Results on large-scale scenes.** Our representation better preserves fine structural details.

with the distance between the sample point and the center of probes, instead of the proposed camera neighborhood; e) “random centers” distributes the light field feature probes by randomly selecting positions from the centers of the input cameras; f) “w/o coordinate scaling  $\nu$ ” disables the scaling of the spherical coordinates by setting the scaling factor  $\nu = (1, 1)$ ; g) “factor concatenation” concatenates the selected factors before the products.

We observe that disabling the core factor  $\bar{V}$  results in a tiny loss of quality in small scene, while disabling core factor  $\bar{M}$ , or the simultaneous removal of both factors leads to a significant drop in performance. Moreover, selecting neighboring factors with the target camera outperforms the “point neighborhood” strategy in the scene with multi-level images, benefiting from our mipmap-like representation. Our factor aggregation model delivers significant performance improvements over the factor concatenation baseline.

## 5. Conclusion

We presented a novel volumetric light field probe representation for high-quality scene reconstruction and rendering. We have also proposed a VMM factorization technique that

	Room			Scuol		
	PSNR↑	SSIM↑	LPIPS ↓	PSNR↑	SSIM↑	LPIPS ↓
full model	30.20	0.871	0.367	27.84	0.855	0.368
a) w/o core factor $\bar{V}$	29.74	0.864	0.382	27.59	0.846	0.382
b) w/o core factor $\bar{M}$	28.16	0.818	0.434	24.55	0.741	0.518
c) w/o core factors $\bar{V}, \bar{M}$	23.78	0.655	0.550	22.76	0.639	0.595
d) point neighborhood	28.97	0.845	0.414	26.72	0.824	0.417
e) random centers	30.24	0.871	0.365	27.81	0.853	0.371
f) w/o coordinate scaling $\nu$	28.65	0.830	0.465	26.97	0.827	0.421
g) factor concatenation	27.68	0.764	0.477	24.78	0.694	0.547

Table 4. **Ablation study** on a 360-degree surrounding room scene and the Scuol scene with free camera trajectories.

decomposes light field probes into products of local factors, resulting in a compact representation with novel view synthesis on par with or better than state-of-the-art methods in diverse scenarios.

## 6. Acknowledgement

Andreas Geiger and Anpei Chen are supported by the ERC Starting Grant LEGO-3D (850533) and DFG EXC number 2064/1 - project number 390727645. We thank Zehao Yu and Binbin Huang for proofreading and their insightful suggestions. We also thank Siyu Tang and Kaifeng Zhao for their valuable discussions and help. Special thanks to **The Flying Dutchman** for the city walk video.



## References

- [1] Sameer Agarwal, Noah Snavely, Ian Simon, Steven M. Seitz, and Richard Szeliski. Building rome in a day. In *ICCV*, 2009. 3
- [2] Alex Yu and Sara Fridovich-Keil, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. *CVPR*, 2022. 1, 3, 7
- [3] Kara-Ali Aliev, Dmitry Ulyanov, and Victor S. Lempitsky. Neural point-based graphics. *arXiv.org*, 1906.08240, 2019. 3
- [4] Benjamin Attal, Jia-Bin Huang, Michael Zollhöfer, Johannes Kopf, and Changil Kim. Learning neural light fields with ray-space embedding networks. In *CVPR*, 2022. 2
- [5] Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *ICCV*, 2021. 7
- [6] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. 2022. 1, 2, 3, 6, 7, 8
- [7] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. *ICCV*, 2023. 2, 3
- [8] Mark Boss, Raphael Braun, Varun Jampani, Jonathan T Barron, Ce Liu, and Hendrik Lensch. Nerf: Neural reflectance decomposition from image collections. In *ICCV*, 2021. 3
- [9] Mark Boss, Varun Jampani, Raphael Braun, Ce Liu, Jonathan Barron, and Hendrik Lensch. Neural-pil: Neural pre-integrated lighting for reflectance decomposition. *Advances in Neural Information Processing Systems*, 2021. 3
- [10] Chris Buehler, Michael Bosse, Leonard McMillan, Steven J. Gortler, and Michael F. Cohen. Unstructured lumigraph rendering. In *ACM Trans. on Graphics*, 2001. 2
- [11] Anpei Chen, Minye Wu, Yingliang Zhang, Nianyi Li, Jie Lu, Shenghua Gao, and Jingyi Yu. Deep surface light fields. In *ACM Trans. on Graphics*, 2018. 2
- [12] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *ECCV*, 2022. 1, 2, 3
- [13] Anpei Chen, Zexiang Xu, Xinyue Wei, Siyu Tang, Hao Su, and Andreas Geiger. Dictionary Fields: Learning a Neural Basis Decomposition. 2023. 5
- [14] Anpei Chen, Zexiang Xu, Xinyue Wei, Siyu Tang, Hao Su, and Andreas Geiger. Factor fields: A unified framework for neural fields and beyond. *arXiv preprint arXiv:2302.01226*, 2023. 5
- [15] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *CVPR*, 2019. 3
- [16] Greg Coombe, Chad Hantak, Anselmo Lastra, and Radek Grzeszczuk. Online construction of surface light fields. In *ESRT*, 2005. 2
- [17] Yuval Eldar, Michael Lindenbaum, Moshe Porat, and Yehoshua Y. Zeevi. The farthest point strategy for progressive image sampling. *TIP*, 1997. 5
- [18] Quankai Gao, Qiangeng Xu, Hao su, Ulrich Neumann, and Zexiang Xu. Strivec: Sparse tri-vector radiance fields. In *ICCV*, 2023. 1
- [19] Steven J Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F Cohen. The lumigraph. In *Seminal Graphics Papers: Pushing the Boundaries*. 2023. 2
- [20] Wenbo Hu, Yuling Wang, Lin Ma, Bangbang Yang, Lin Gao, Xiao Liu, and Yuewen Ma. Tri-miprf: Trimip representation for efficient anti-aliasing neural radiance fields. In *ICCV*, 2023. 2
- [21] Haiyan Jin, Isabella Liu, Peijia Xu, Xiaoshuai Zhang, Songfang Han, Sai Bi, Xiaowei Zhou, Zexiang Xu, and Hao Su. Tensorf: Tensorial inverse rendering. In *CVPR*, 2023. 3
- [22] Petr Kellnhofer, Lars C Jebe, Andrew Jones, Ryan Spicer, Kari Pulli, and Gordon Wetzstein. Neural lumigraph rendering. In *CVPR*, 2021. 2
- [23] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. on Graphics*, 2023. 3, 8
- [24] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 6
- [25] Abhijit Kundu, Kyle Genova, Xiaoqi Yin, Alireza Fathi, Caroline Pantofaru, Leonidas Guibas, Andrea Tagliasacchi, Frank Dellaert, and Thomas Funkhouser. Panoptic Neural Fields: A Semantic Object-Aware Neural Scene Representation. In *CVPR*, 2022. 7
- [26] Florent Lafarge and Clément Mallet. Creating large-scale city models from 3d-point clouds: a robust approach with hybrid representation. *International journal of computer vision*, 2012. 3
- [27] Marc Levoy and Pat Hanrahan. Light field rendering. In *ACM Trans. on Graphics*, pages 31–42, 1996. 2
- [28] Zhong Li, Liangchen Song, Celong Liu, Junsong Yuan, and Yi Xu. Neulf: Efficient novel view synthesis with neural 4d light field. *arXiv preprint arXiv:2105.07112*, 2021. 2
- [29] Yiyi Liao, Jun Xie, and Andreas Geiger. Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *PAMI*, 2022. 2, 6, 7
- [30] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. In *NeurIPS*, 2020. 3

- [31] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. In *ACM Trans. on Graphics*, 2019. 3
- [32] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. *arXiv.org*, 2008.02268, 2020. 3
- [33] Morgan McGuire, Mike Mara, Derek Nowrouzezahrai, and David Luebke. Real-time global illumination using precomputed light field probes. 2017. 3
- [34] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *CVPR*, 2019. 3
- [35] Andreas Meuleman, Yu-Lun Liu, Chen Gao, Jia-Bin Huang, Changil Kim, Min H. Kim, and Johannes Kopf. Progressively optimized local radiance fields for robust view synthesis. In *CVPR*, 2023. 1, 3
- [36] Zhenxing Mi and Dan Xu. Switch-nerf: Learning scene decomposition with mixture of experts for large-scale neural radiance fields. In *ICLR*, 2023. 3
- [37] Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: practical view synthesis with prescriptive sampling guidelines. In *ACM Trans. on Graphics*, 2019. 2
- [38] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 1, 3, 4, 7
- [39] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. on Graphics*, 2022. 1, 3, 6, 7
- [40] Michael Niemeyer, Lars M. Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *CVPR*, 2020. 3
- [41] Michael Oechsle, Michael Niemeyer, Lars Mescheder, Thilo Strauss, and Andreas Geiger. Learning implicit surface light fields. *arXiv.org*, 2003.12406, 2020. 2
- [42] Jeong Joon Park, Peter Florence, Julian Straub, Richard A. Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *CVPR*, 2019. 3
- [43] Ravi Ramamoorthi. Nerfs: The search for the best 3d representation. *arXiv.org*, 2023. 3
- [44] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *ICCV*, 2021. 1
- [45] Ari Silvennoinen and Jaakko Lehtinen. Real-time global illumination by precomputed local reconstruction from sparse radiance probes. *ACM Trans. on Graphics*, 2017. 3
- [46] Vincent Sitzmann, Semon Rezchikov, William T. Freeman, Joshua B. Tenenbaum, and Fredo Durand. Light field networks: Neural scene representations with single-evaluation rendering. In *NeurIPS*, 2021. 2
- [47] Mohammed Suhail, Carlos Esteves, Leonid Sigal, and Ameesh Makadia. Light field neural rendering. In *CVPR*, 2022. 2
- [48] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. *CVPR*, 2022. 1, 3, 7
- [49] Matthew Tancik, Vincent Casser, Xinchun Yan, Sabeek Pradhan, Ben P. Mildenhall, Pratul P. Srinivasan, Jonathan T. Barron, and Henrik Kretzschmar. Block-nerf: Scalable large scene neural view synthesis. In *CVPR*, 2022. 1, 3
- [50] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Justin Kerr, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, David McAllister, and Angjoo Kanazawa. Nerfstudio: A modular framework for neural radiance field development. In *ACM Trans. on Graphics*, 2023. 6
- [51] Ayush Tewari, Justus Thies, Ben Mildenhall, Pratul Srinivasan, Edgar Tretschk, Wang Yifan, Christoph Lassner, Vincent Sitzmann, Ricardo Martin-Brualla, Stephen Lombardi, et al. Advances in neural rendering. In *Computer Graphics Forum*, 2022. 3
- [52] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: image synthesis using neural textures. *ACM Trans. on Graphics*, 2019. 3
- [53] Ledyard R Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 1966. 5
- [54] Haithem Turki, Deva Ramanan, and Mahadev Satyanarayanan. Mega-nerf: Scalable construction of large-scale nerfs for virtual fly-throughs. In *CVPR*, 2022. 1, 3
- [55] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T. Barron, and Pratul P. Srinivasan. Ref-NeRF: Structured view-dependent appearance for neural radiance fields. 2022. 3
- [56] Huan Wang, Jian Ren, Zeng Huang, Kyle Olszewski, Menglei Chai, Yun Fu, and Sergey Tulyakov. R2L: Distilling neural radiance field to neural light field for

- efficient novel view synthesis. In *European Conference on Computer Vision*, pages 612–629. Springer, 2022. 2
- [57] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In *NeurIPS*, 2021. 3
- [58] Peng Wang, Yuan Liu, Zhaoxi Chen, Lingjie Liu, Ziwei Liu, Taku Komura, Christian Theobalt, and Wenping Wang. F2-nerf: Fast neural radiance field training with free camera trajectories. 2023. 1, 2, 3, 6, 7, 8
- [59] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P. Srinivasan, Howard Zhou, Jonathan T. Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas A. Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *CVPR*, 2021. 2, 4
- [60] Suttisak Wizatwongsa, Pakkapon Phongthawee, Jiraphon Yenphraphai, and Supasorn Suwajanakorn. Nex: Real-time view synthesis with neural basis expansion. In *CVPR*, 2021. 2
- [61] Daniel N. Wood, Daniel I. Azuma, Ken Aldinger, Brian Curless, Tom Duchamp, David Salesin, and Werner Stuetzle. Surface light fields for 3d photography. In *ACM Trans. on Graphics*, 2000. 2
- [62] Yuanbo Xiangli, Linning Xu, Xingang Pan, Nanxuan Zhao, Anyi Rao, Christian Theobalt, Bo Dai, and Dahua Lin. Bungeenerf: Progressive neural radiance field for extreme multi-scale scene rendering. In *ECCV*, 2022. 1, 2, 3, 6
- [63] Linning Xu, Yuanbo Xiangli, Sida Peng, Xingang Pan, Nanxuan Zhao, Christian Theobalt, Bo Dai, and Dahua Lin. Grid-guided neural radiance fields for large urban scenes. In *CVPR*, 2023. 1, 3, 8
- [64] Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-nerf: Point-based neural radiance fields. In *CVPR*, 2022. 3
- [65] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. In *NeurIPS*, 2021. 3
- [66] Zehao Yu, Anpei Chen, Bozidar Antic, Songyou Peng, Apratim Bhattacharyya, Michael Niemeyer, Siyu Tang, Torsten Sattler, and Andreas Geiger. Sdfs-tudio: A unified framework for surface reconstruction, 2022. 6
- [67] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. In *NeurIPS*, 2022. 3
- [68] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv.org*, 2010.07492, 2020. 1, 7
- [69] Kai Zhang, Fujun Luan, Qianqian Wang, Kavita Bala, and Noah Snavely. Physg: Inverse rendering with spherical gaussians for physics-based material editing and relighting. In *CVPR*, 2021. 3
- [70] Yuanqing Zhang, Jiaming Sun, Xingyi He, Huan Fu, Rongfei Jia, and Xiaowei Zhou. Modeling indirect illumination for inverse rendering. In *CVPR*, 2022. 3
- [71] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: learning view synthesis using multiplane images. *ACM Trans. on Graphics*, 2018. 2, 3