Eurographics Symposium on Rendering (2022)
A. Ghosh and L.-Y. Wei (Editors)

NeuLF: Efficient Novel View Synthesis with Neural 4D Light Field

Zhong Li[1] and Liangchen Song[2†] and Celong Liu[1] and Junsong Yuan[2] and Yi Xu[1]

[1]OPPO US Research Center, InnoPeak Technology, Inc, Palo Alto, California, USA
[2]University at Buffalo, State University of New York, Buffalo, New York, USA

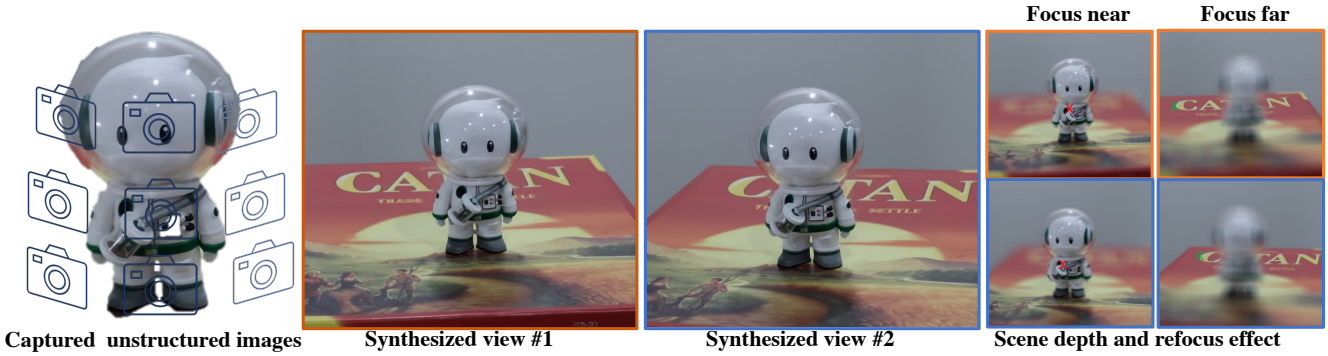

Figure 1: *Given a set of images captured in front of a scene using a handheld camera,* ***Neural 4D Light Field (NeuLF)*** *uses an implicit neural representation to learn the mapping from rays to color values. With the learned model, novel views can be synthesized by predicting the color of each ray. Moreover, with our framework, auto-refocus effect can be inherently generated.*

Abstract

In this paper, we present an efficient and robust deep learning solution for novel view synthesis of complex scenes. In our approach, a 3D scene is represented as a light field, i.e., a set of rays, each of which has a corresponding color when reaching the image plane. For efficient novel view rendering, we adopt a two-plane parameterization of the light field, where each ray is characterized by a 4D parameter. We then formulate the light field as a function that indexes rays to corresponding color values. We train a deep fully connected network to optimize this implicit function and memorize the 3D scene. Then, the scene-specific model is used to synthesize novel views. Different from previous light field approaches which require dense view sampling to reliably render novel views, our method can render novel views by sampling rays and querying the color for each ray from the network directly, thus enabling high-quality light field rendering with a sparser set of training images. Per-ray depth can be optionally predicted by the network, thus enabling applications such as auto refocus. Our novel view synthesis results are comparable to the state-of-the-arts, and even superior in some challenging scenes with refraction and reflection. We achieve this while maintaining an interactive frame rate and a small memory footprint.

CCS Concepts

• ***Computing methodologies*** → *Rendering; Computer vision problems; Virtual reality;*

1. Introduction

Novel view synthesis has long been studied by the computer vision and computer graphics community. It has many applications in multimedia, AR/VR, gaming, etc. Traditional computer vision approaches such as multi-view stereo (MVS) and structure-from-motion (SfM) aim to build a geometric representation of the scene first. An alternative approach is image-based rendering [LH96, GGSC96, BBM*01], where no underlying geometric model or only a simple proxy is needed. These methods can achieve photorealistic rendering. However, a typical light field setup prefers a dense sampling of views around a scene, which limits practical use of such methods.

Thanks to the recent advancement of neural rendering [TFT*20],

† This work was done while Liangchen Song was an intern at OPPO.

DOI: 10.2312/sr.20221156

photorealistic rendering with only a sparse set of inputs can be achieved. One approach is to use an explicit geometric representation of a scene reconstructed using a traditional computer vision pipeline and learning-based rendering. Object-specific or category-specific meshes or multi-plane images (MPI) [ZTF*18] can be used as the representation. However, these explicit representations do not allow a network to learn the optimal representation of the scene. To achieve this, volume-based representations can be used [**?**]. But they typically require a large amount of memory space, especially for complex scenes.

Memory-efficient implicit representations have gained interests from the research community. For example, surface-based implicit representations can achieve state-of-the-art results and can provide a high-quality reconstruction of the scene geometry [KJJ*21]. However, surface-based representations face challenges when dealing with complex lighting and geometry, such as transparency, translucency, and thin geometric structures. More recently, volume-based implicit representation achieves remarkable rendering results (e.g., NeRF [MST*20]) and inspires follow-up research. One drawback of NeRF, nevertheless, is the time complexity of rendering. NeRF, in its original form, needs to query the network multiple times per ray and accumulate color and density along the query ray, which prohibits real-time applications. Although there have been many efforts to accelerate NeRF, they typically require depth proxy to train or rely on additional storage to achieve faster rendering [HSM*21, YLT*21, RPLG21].

We propose an efficient novel view synthesis framework, which we call Neural 4D Light Field (NeuLF). We define a scene as an implicit function that maps 4D light field rays to corresponding color values directly. This function can be implemented as a Multilayer Perceptron (MLP) and can be composed using only a sparse set of calibrated images placed in front of the scene. This formulation allows the color of a camera ray to be learned directly by the network and does not require a time-consuming ray-marcher during rendering as in NeRF. Thus, NeuLF achieves hundreds times speedup over NeRF during inference, while producing similar or even better rendering quality. Our light field setup limits the novel viewpoints to be on the same side of the cameras, e.g., front views only. Despite these constraints, we argue that for many applications such as teleconferencing, these are reasonable trade-offs to gain much faster inference speed with high-quality rendering and a small memory footprint.

Comparison with NeRF: Although our work is inspired by NeRF [MST*20], there are some key distinctions. NeRF represents the continuous scene function as a 5D radiance field. Such a representation bring redundancy, i.e., color along a ray is constant in free space. By restricting the novel viewpoints to be outside of the convex hull of the object, the 5D radiance field can be reduced to a light field in a lower dimension, e.g. 4D. Table 1 summarizes the differences between NeRF and NeuLF.

Moreover, NeuLF can also optionally estimate per ray depth by enforcing multi-view and depth consistency. Leveraging the depth information, applications such as auto refocus can be enabled. We show state-of-the-art novel view synthesis results on benchmark datasets and our own captured data (Fig. 1). The comparisons with existing approaches also validate the efficiency and effectiveness of our proposed method. In summary, our contributions are:

Table 1: *The comparisons between our proposed NeuLF and NeRF [MST*20].*

	NeRF	NeuLF
Input	5*D*	4*D*
Output	radiance, density	color
Viewpoint range	360°	front views
Rendering method	raymarching	direct evalution
Rendering speed	slow	fast
Memory consumption	small	small
High-quality rendering	yes	yes

- We proposed a fast and memory-efficient novel view synthesis pipeline, which solves the mapping from 4D rays to colors directly.
- Compared with the state-of-the-arts, our method is better than NeRF [MST*20] and NeX [WPYS21] when the scene contains challenging refraction and reflection effects. In addition, our method only needs 25% of the original input on those challenge scenes to achieve a similar or even better quality.
- Application wise, the framework we proposed can optionally estimate depth per ray; thus enabling applications such as 3D reconstruction and auto refocus.

2. Related Work

Our work builds upon previous work in traditional image-based rendering and implicit-function-based neural scene representation. In the following sections, we will review these fields and beyond in detail.

Image-based Rendering: For novel view synthesis, image-based rendering has been studied as an alternative to geometric methods. In the seminal work of light field rendering [LH96, GGSC96], a 5D radiance field is reduced to a 4D light field considering the radiance along a ray remains constant in free space [SRF*21, FV21, SESM22, AHZ*22]. The ray set in a light field can be parameterized in different ways, among which two-plane parameterization is the most common one. Rendering novel views from the light field involves extracting corresponding 2D slices from the 4D light field. To achieve better view interpolation, approximate geometry can be used [GGSC96, BBM*01, WAA*00, ZLY*17]. Visual effects of variable focus and variable depth-of-field can also be achieved using light field [IMG00].

With the advancement of deep learning, a few learning-based methods have been proposed to improve the traditional light field. For example, LFGAN [CRL20] can learn texture and geometry information from light field data sets and in turn predict a small light field from one RGB image. Meng et al. [MWLL20] enables high-quality reconstruction of a light field by learning the geometric features hierarchically using a residual network. [WLFC21] integrates an anti-aliasing module in a network to reduce the artifacts in the reconstructed light field. Our method learns an implicit function of the light field and achieves high-quality reconstruction with a sparse input.

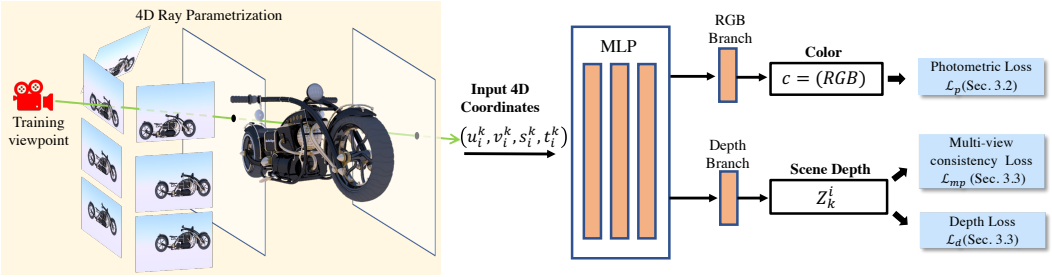

Figure 2: *An overview of the Neural 4D Light Field (NeuLF). For a set of sampled rays from training images, their 4D coordinates and the corresponding color values can be obtained. The input for NeuLF is the 4D coordinate of a ray (query) and the output is its RGB color and scene depth. By optimizing the differences between the predicted colors and ground-truth colors, NeuLF can faithfully learn the mapping between a 4D coordinate that characterizes the ray and its color. We also build a depth branch to let the network learn the per ray scene depth by self-supervised losses $\mathcal{L}_{mp}$ and $\mathcal{L}_d$.*

Neural Scene Representation: Neural rendering is an emerging field. One of the most important applications of neural rendering is novel view synthesis. A comprehensive survey of the topic can be found in [TFT*20].

An explicit geometric model can be used as the representation of a scene. [RK20] creates a proxy geometry of the scene using SfM and MVS. Then, a recurrent encoder-decoder network is used to synthesize new views from nearby views. To improve blending on imperfect meshes from MVS, [HPP*18] uses predicted weights from a network to perform blending. A high-quality parameterized mesh of the human body [ZFT*21] and category-specific mesh reconstruction [KTEM18] can also be used as the proxy. Recently, Multi-plane Image (MPI) [ZTF*18] has gained popularity. [ZTF*18] learns to predict MPIs from stereo images. The range of novel views is later improved by [STB*19]. [FBD*19] uses learned gradient descent to generate an MPI from a set of sparse inputs. [MSOC*19] uses an MPI representation for turning each sampled view into a local light field. NeX [WPYS21] represents each pixel of an MPI with a linear combination of basis functions and achieves state-of-the-art rendering results in real-time. MPI representation might typically lead to stack-of-cards artifacts. Both NeX and NeuLF compute ray-plane intersections. However, NeX is a volumetric rendering method that integrates at multiple intersections on the planes, which are used to store information of the scene. NeuLF is a light field representation that uses the coordinates of ray-plane intersections as an input to mapping a color value. [STH*19] trains a network to reconstruct both the geometry and appearance of a scene on a 3D grid. For dynamic scenes, Neural Volumes (NV) [**?**] uses an encoder-decoder network to convert input images into a 3D volume representation. [LSS*21] extends NV using a mixture of volumetric primitives to achieve better and faster rendering. While volume-based representations allow for learning the 3D structure, they require large memory space, especially for large scenes.

Implicit-function-based approaches provide memory-efficient alternatives to explicit representations, while still allowing learning the 3D structure of the scene. Implicit representations can be categorized as implicit surface-based and implicit volume-based approaches. SRN [SZW19] maps 3D coordinates to a local feature embedding at these coordinates. Then, a trained ray-marcher and a pixel generator are used to render novel views. IDR [YKM*20] uses an implicit Signed Distance Function (SDF) to model an object on 3D surface reconstruction. Neural Lumigraph [KJJ*21] provides even better rendering quality by utilizing a sinusoidal representation network (SIREN) to model the SDF.

Our work is inspired by NeRF [MST*20], which uses a network to map continuous 5D coordinates (location and view direction) to volume density and view-dependent radiance. Recent works have extended NeRF to support novel illumination conditions [SDZ*21], rendering from unstructured image collections from the internet [MBRS*21], large-scale unbounded scenes [ZRSK20], unknown camera parameters [WWX*21], anti-aliasing [BMT*21], deformable models [PSB*21], dynamic scenes [**?**], etc. A lot of effort has been put into speeding up rendering with NeRF. DONeRF [NSP*21] places samples around scene surfaces by predicting sample locations along each ray. However, transparent objects will pose issues and it requires ground-truth depth for training. FastNeRF [GKJ*21] achieves 200fps by factoring NeRF into a position-dependent network and a view-dependent network. This allows efficient caching of network outputs during rendering. [YLT*21] trains a NeRF-SH network, which maps coordinates to spherical harmonic coefficients and pre-samples the NeRF-SH into a sparse voxel-based octree structure. These pre-sampling approaches sacrifice additional memory storage for speedups. NSVF [LGL*20] represents a scene using a set of NeRF-like implicit fields defined on voxels and uses a sparse octree to achieve 10x speedup over NeRF during rendering. KiloNeRF [RPLG21] decomposes a scene into a grid of voxels and uses a smaller NeRF for each voxel. Storage costs will increase when more networks are used. Using AutoInt [LMW21], calculations of any definite integral can be done in two network evaluations; this achieves 10x acceleration, but rendering quality is decreased. Compared with these approaches, our method achieves hundreds times speedup over NeRF by representing the scene with an implicit 4D light field without any additional pre-sampling or storage overhead.

Recently LFN [SRF*21] propose to use a network to direct

regress the mapping from the 6D Plücker coordinates to colors. It leverages meta-learning to enables view synthesis using a single image observation in ShapeNet dataset [CFG*15]. In contrast, we use 4D representation and conduct extensive experiments on real-world scenes. A concurrent work [FV21] transforms 4D light field representation by leveraging Gegenbauer polynomials basis, and learns the mapping from this basis function to color. However, it requires dense narrow baseline input with planar camera arrangement. LFNR [SESM22] introduces a two-stage transformer-based model that maps the 4D representation ray to color. They produce reference view features along the epipolar lines, then align features cross views to produce the ray color. They achieve the state-of-the-art results on the various datasets. However, their transformer-based two-stage network is computationally expensive in its current form.

3. Our Method

In Fig. 2, we illustrate the pipeline of our system. In the following sections, we will first briefly discuss the light field, followed by our NeuLF representation and the proposed loss functions. We will also discuss our training strategies.

3.1. 4D Light Field Representation

All possible light rays in a space can be described by a 5D plenoptic function. Since radiance along a ray is constant if viewed from outside of the convex hull of the scene, this 5D function can be reduced to a 4D light field [LH96, GGSC96]. The most common parameterization is a two-plane model shown in Fig. 3. Each ray from the camera to the scene will intersect with the two planes. Thus, we can represent each ray using the coordinates of the intersections, (u,v) and (s,t), or simply a 4D coordinate (u,v,s,t). Using this representation, all rays from the object to one side of the two planes can be uniquely determined by a 4D coordinate.

Based on this representation, rendering a novel view can be done by querying all the rays from the center of projection to every pixel on the camera's image plane. We denote them as $\{R_1, R_2, ..., R_N\}$, where N is the total number of pixels. Then, for the i-th ray R_i, we can obtain its 4D coordinate (u_i, v_i, s_i, t_i) by computing its intersections with the two planes. If a function f maps the continuous 4D coordinates to color values, we can obtain the color of R_i by

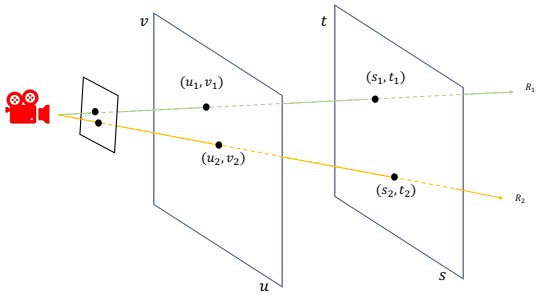

Figure 3: *The 4D light field representation. Each ray is characterized by 4 parameters (u,v) and (s,t), which uniquely locate the ray.*

evaluating the function $f_c(u_i, v_i, s_i, t_i)$. In the next section, we will introduce Neural 4D Light Field (NeuLF) for reconstructing this mapping function f.

3.2. Neural 4D Light Field Reconstruction

We formulate the mapping function f_c as a Multilayer Perceptron (MLP). The input of this MLP is a 4D coordinate and the output is RGB color. As shown in Fig. 2, the goal of the network is to learn the mapping function from training data.

Training Data: for a given scene, the training data come from a set of captured images $\{I_1, I_2, ..., I_M\}$, where M is the total number of images. Assuming the camera pose for each image is known or obtainable, for each image $I_k (k = 1, ..., M)$, we can traverse its pixels and generate all corresponding rays $\{R_1^k, R_2^k, ..., R_{N_k}^k\}$, where N_k is the total number of pixels in the k-th image. Based on the 4D light field representation, all 4D coordinates $\left\{\left(u_1^k, v_1^k, s_1^k, t_1^k\right), ..., \left(u_{N_k}^k, v_{N_k}^k, s_{N_k}^k, t_{N_k}^k\right)\right\}, (k = 1, ..., M)$, can be obtained. On the other hand, the color for each pixel is known from the input images. To this end, we have constructed a collection of sample mappings from 4D coordinates to color values $\left(u_i^k, v_i^k, s_i^k, t_i^k\right) \rightarrow c_i^k, (k = 1...M, i = 1...N_k)$, where c_i^k is the color of the i-th pixel on the k-th image. By feeding this training data to the MLP network, the parameters Θ can be learned by minimizing the following photometric loss $\mathcal{L}_p$:

$$\mathcal{L}_p = \sum_{k=1}^{M} \sum_{i=1}^{N_k} \left\| f_c\left(u_i^k, v_i^k, s_i^k, t_i^k \mid \Theta\right) - c_i^k \right\|_2 \tag{1}$$

In Fig. 2, we demonstrate an example of capturing images to train our neural 4D light field representation with a set of unstructured front-faced camera views. In this example, the cameras are placed on one side of the two light slabs.

Rendering: Given a viewpoint $\mathcal{V}$, we can render a novel view $\mathcal{R}(\mathcal{V})$ by evaluating the learned mapping function f. With the camera pose and the desired rendering resolution $\{W^{\mathcal{V}}, H^{\mathcal{V}}\}$, we sample all rays $\{R_1^{\mathcal{V}}, R_2^{\mathcal{V}}, ..., R_{N_{\mathcal{V}}}^{\mathcal{V}}\}$, where $N_{\mathcal{V}} = W^{\mathcal{V}} \times H^{\mathcal{V}}$ is the number of pixels to be rendered. We can further calculate the 4D coordinates $\left\{\left(u_i^{\mathcal{V}}, v_i^{\mathcal{V}}, s_i^{\mathcal{V}}, t_i^{\mathcal{V}}\right)\right\}$ for each ray $R_i^{\mathcal{V}}, (i = 1...N_{\mathcal{V}})$. We then formulate the rendering process $\mathcal{R}$ as $N_{\mathcal{V}}$ evaluations of the mapping function f_c:

$$\mathcal{R}(\mathcal{V}) = \left\{ f_c\left(u_i^{\mathcal{V}}, v_i^{\mathcal{V}}, s_i^{\mathcal{V}}, t_i^{\mathcal{V}} \mid \Theta\right), i = 1, ..., N_{\mathcal{V}} \right\}. \tag{2}$$

3.3. Scene Depth Estimation

To simulate variable focus and variable depth-of-field using light field, previous work [IMG00] dynamically reparameterizes the light field by manually moving the focal surface as a plane. To optimally and automatically select a focal plane given a pixel location, we aim to solve the scene geometry under Lambertian scene assumption.

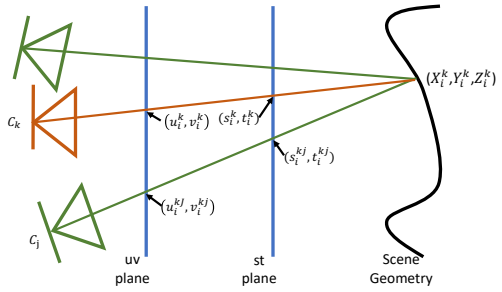

Figure 4: *Given the 4D query ray $(u_i^k, v_i^k, s_i^k, t_i^k)$ (red line) and its predicted depth Z_i^k, we compute its corresponding rays (green line) to its nearest camera views C_j, which are used for self-supervision.*

We let the same MLP predict per-ray depth as shown in Fig. 2. As in Fig. 4, for a 4D query ray $(u_i^k, v_i^k, s_i^k, t_i^k)$ from the camera C_k, the predicted scene depth is $Z_i^k = f_d(u_i^k, v_i^k, s_i^k, t_i^k \mid \Theta)$ where f_d is the network that maps from 4D coordinates to depth. We compute ray-surface intersection (X_i^k, Y_i^k, Z_i^k). Then, we self-supervise Z_i^k by applying multi-view consistency cues. To achieve this, we trace the rays from (X_i^k, Y_i^k, Z_i^k) back to C_k's K-nearest data cameras $\{C_j\}, (j = 1, ..., K)$ (K=5 in our experiments). Those rays intersect *uv-st* planes and can be parameterized as: $\{u_i^{kj}, v_i^{kj}, s_i^{kj}, t_i^{kj}\}$

Ray-plane intersection is differentiable. We thus propose two loss functions $\mathcal{L}_{mp}$ and $\mathcal{L}_d$ to minimize the multi-view photometric error and depth differences respectively as follows:

$$\mathcal{L}_{mp} = \sum_{k=1}^{M} \sum_{i=1}^{N_k} \left\| f_c\left(u_i^k, v_i^k, s_i^k, t_i^k \mid \Theta\right) - \mathcal{C}_i^k(Z_i^k) \right\|_2 \tag{3}$$

$$\mathcal{C}_i^k(Z_i^k) = \sum_{j=1}^{K} \omega(i,j) f_c\left(u_i^{kj}, v_i^{kj}, s_i^{kj}, t_i^{kj} \mid \Theta\right) \tag{4}$$

$$\mathcal{L}_d = \sum_{k=1}^{M} \sum_{i=1}^{N_k} \left\| f_d\left(u_i^k, v_i^k, s_i^k, t_i^k \mid \Theta\right) - \mathcal{D}_i^k(Z_i^k) \right\|_2 \tag{5}$$

$$\mathcal{D}_i^k(Z_i^k) = \sum_{j=1}^{K} \omega(i,j) f_d\left(u_i^{kj}, v_i^{kj}, s_i^{kj}, t_i^{kj} \mid \Theta\right) \tag{6}$$

where $\omega(i,j) = \frac{\frac{1}{d_{ij}^2}}{\sum_j \frac{1}{d_{ij}^2}}$ is the normalized weights with the Euclidean distance between the camera i and its neighbor camera j as d_{ij}. $\mathcal{C}_i^k(Z_i^k)$ and $\mathcal{D}_i^k(Z_i^k)$ are the color and depth summed over weighted nearest data cameras. Training with both $\mathcal{L}_{mp}$ and $\mathcal{L}_d$ will encourage the MLP to learn the depth representation of the scene.

With ray-based depth, we can enable efficient auto-refocus effect by adopting a dynamic parameterization of the light field [IMG00]. More specifically, we simulate the depth-of-field effect by combining rays within an given aperture size. Fig. 5 shows the case of two

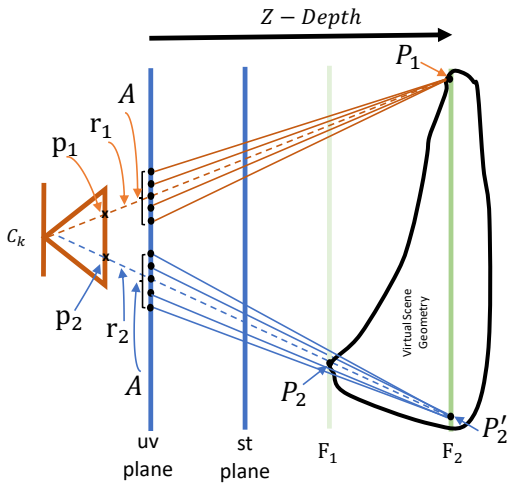

Figure 5: *Auto-refocus illustration. Reconstruct two rays r_1 and r_2 by combining ray's within their aperture A. When focal plane correspond to r_1's depth P_1, the r_1 will appears in focus, while r_2 will not.*

rays r_1 and r_2. The two rays intersect camera image plane at pixels p_1 and p_2, intersect the scene geometry at P_1 and P_2, and intersect a given focal plane F_2 at P_1 and P_2'. To reconstruct the final color of the pixels p_1 and p_2, we collect a cone of sample rays originated from P_1 and P_2' on the focal plane F_2 within an aperture A. We then query the network f_c to obtain the ray colors. These ray colors are weighted-averaged to produce the final pixel color. In this case, P_1 is on the surface of the object, while P_2' is not. Thus, the image pixel p_1 appears in focus, while pixel p_2 is blurred since it combines colors from a small area around the actual surface point P_2.

To auto-refocus at pixel location p_2, we extract its depth by query the depth network, and set a new focal plane F_1 at pixel p_2's depth. Rendering the NeuLF with this new focal plane will make pixel p_2 in focus while blurring pixel $p1$.The results are shown in Fig. 6.

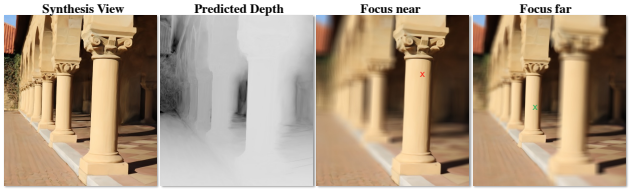

Figure 6: *Depth estimation results and refocus effects. We show a novel view and its corresponding depth estimation. In addition, we show two auto-refocus results with focus set on the red "x" (near) and on the green "x" (far) given by a user click.*

4. Experimental Results

We first discuss the implementation details of NeuLF. Then we perform quantitative and qualitative evaluations against state-of-the-art methods for novel view synthesis.

4.1. Implementation Details

We train the MLP on the following overall loss function:

$$\mathcal{L} = \mathcal{L}_p + \lambda_s \mathcal{L}_{mp} + \lambda_r \mathcal{L}_d, \tag{7}$$

where the weighting coefficients are $\lambda_s = 0.5$ and $\lambda_r = 0.1$. These parameters are fine-tuned by mixing the manual tuning and grid search tuning. This set of parameters works best for most of the scenes we tested. For scenes with strong view-dependent effects such as specularities, Lambertian assumption is no longer valid. Therefore, we disable the multi-view photometric error and depth difference terms in the loss for such scenes. Grid search can be used to automatically find optimal parameters for a specific scene but will lead to a longer training time.

To extract camera rays from input photos, we calibrate the camera poses and intrinsic parameters using a structure-from-motion tool from COLMAP [SF16]. During training, we randomly select a batch of camera rays from the training set at each iteration. By passing them to the MLP to predict the color of each ray, we calculate the loss and back-propagate the error.

The input 4D coordinate (u,v,s,t) (normalized to $[-1,1]$) is passed through 20 fully-connected ReLU layers, each with 256 channels. Our structure includes a skip connection that concatenates the input 4D coordinate to every 4 layers start with the fifth layer. An additional layer outputs 256-dimensional feature vector. This feature vector is split into the color branch and depth branch. Each branch is followed with an additional fully-connected ReLU layer with 128 channels, and outputs 3 channel RGB radiance with sigmoid activation and 1 channel scene depth with sigmoid activation, respectively. For model training, we set the ray batch size in each iteration to $8,192$. We train the MLP for 1,500 epochs using the Adam optimizer [KB14]. The initial learning rate is 5×10^{-4} and decays by 0.995 every epoch. To train the NeuLF on a scene with 30 input images with a resolution of 567×1008, it takes 5 hours using 1 Nvidia RTX 3090 card. For testing on the same situation, rendering an image costs about 70ms while NeRF takes 51 seconds.

4.2. Comparison with State-of-the-Art Methods

In this section, we demonstrate qualitative results of novel view synthesis and compare them with current top-performing approaches: NeRF [MST*20] and NeX [WPYS21] as well as the baseline light field rendering method [LH96]. We evaluate the models on the shinny dataset [WPYS21]. In this dataset, each scene is captured by a handheld smartphone in a forward-facing manner with a resolution 567×1008 or 756×1008. This is a challenging dataset that contains complex scene geometry and various challenging view-dependent effects. (e.g.,, refraction through the test tubes filled with liquid and magnifier, rainbow effect emitted by a CD disk, and sharp specular highlights from silverware and thick glasses.)

We hold out $\frac{1}{8}$ of each scene as the test set and use the rest of them as the training set. The qualitative results are shown in Fig. 7. The leftmost column shows our results on the test view of three challenging scenes (Lab, CD, and Tools). We have zoomed in on parts of the image areas for comparison with other methods. Our method is superior when a scene contains detailed refraction and reflection. In the Lab scene, the metal frames behind the tubes are faithfully recovered by our methods. In the CD scene, our result produce more sharp and vivid detail on the rainbow, and less noise on the liquid bottle than NeX and NeRF. In the Tools scene, although our result is not as sharp as the ground truth, it contains more overall details than NeX and NeRF and is able to recover metallic reflection with less noise than others. Our method essentially relies on ray interpolation rather than volumetric rendering like NeRF/NeX. We believe the traditional ray interpolation handles refraction and reflection better than volumetric representation.

The baseline Light Field rendering (last column) exhibits good results when rays are sufficiently sampled (magnifier). However, it exhibits aliasing and misalignment artifacts in the low sampling area (metallic, tube).

We report three metrics: PSNR (Peak Signal-to-Noise Ratio, higher is better), SSIM (Structural Similarity Index Measure, higher is better), and LPIPS (learned perceptual Image Patch Similarity, lower is better) to evaluate our test results. In Tab 2, we report the three metrics for the 8 scenes in the shinny dataset. We use the NeX and NeRF scores originally reported in the NeX paper. For each scene, we calculate the scores by averaging across the views in the test split. Our method produces the highest score across all three metrics on CD and Lab scenes which contain challenging refraction and reflection. For the rest scenes, while NeX has the highest score by producing high-frequency details in the richly-textured area, we generate comparable scores as NeRF. Note that our rendering speed is hundreds times faster than NeRF.

4.3. Ablation Study

To demonstrate the effectiveness of our proposed multi-view and depth regularization terms(MVDL), we use two mostly Lambertian scenes the *columns* and the *tribe*, which contain complex local details, for ablation studies. We demonstrate how the multi-view and depth loss $\mathcal{L}_{mp}$ and $\mathcal{L}_d$ affect the quality of synthesized views. We also compare the model trained with and without our proposed regularization terms. As shown in Tab. 8 and Fig. 3, our model with multi-view and depth regularization improves the results both qualitatively and quantitatively in both scenes, of which more details can be reconstructed. For Lambertian scenes, the network trained with multi-view and depth consistency constraints takes the scene geometry into account. Hence the results in unseen views are more reasonable and detail preserved.

4.4. Study on Number of Inputs

To understand how the number of input views affect the novel view synthesis result, we train our model on fewer images. As shown in

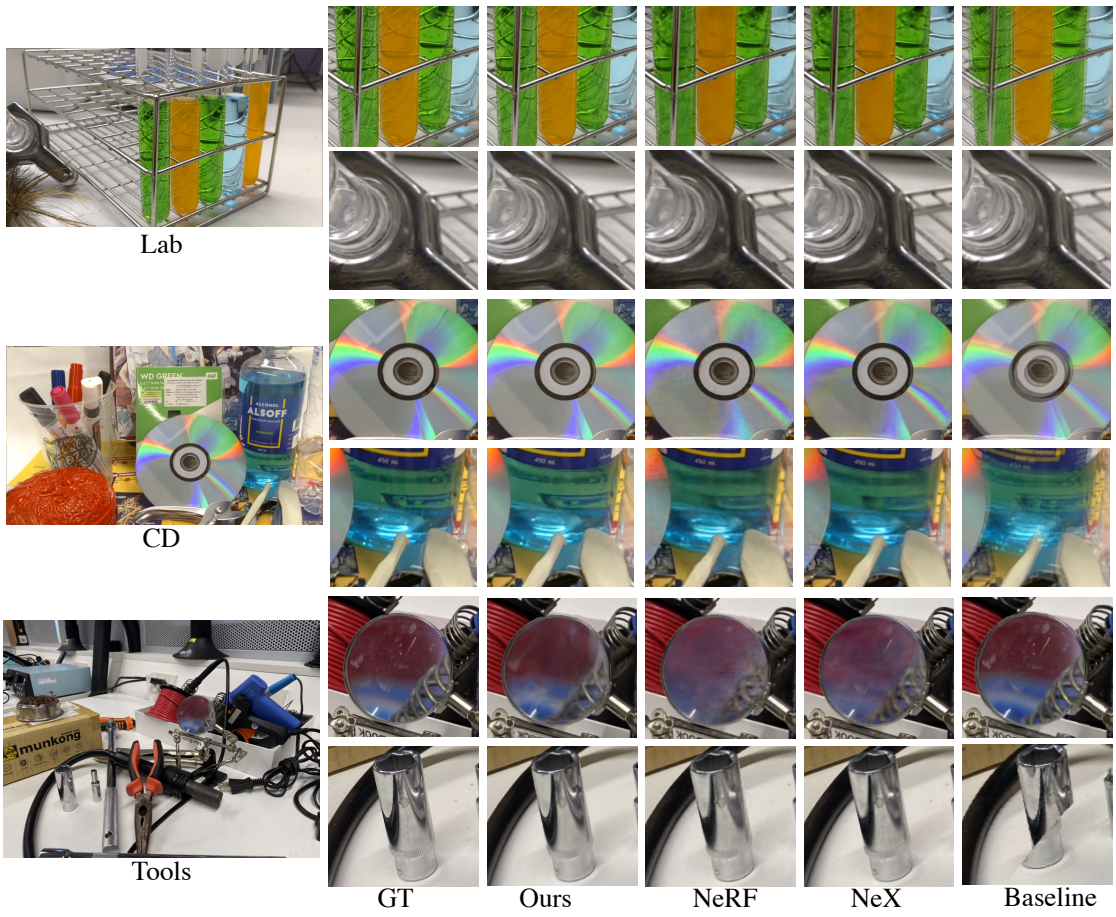

Figure 7: *Qualitative Results on test views from shinny dataset. Our method captures more details on the reflection and refraction areas of the scenes.*

Table 2: *Average scores across test views for each scene in the shinny dataset.*

	PSNR↑				SSIM↑				LPIPS↓			
	Baseline	NeX	NeRF	Ours	Baseline	NeX	NeRF	Ours	Baseline	NeX	NeRF	Ours
Lab	21.69	30.43	29.60	**31.95**	0.693	0.949	0.936	**0.951**	0.261	0.146	0.182	**0.097**
CD	20.70	31.43	30.14	**32.11**	0.551	0.958	0.937	**0.964**	0.294	0.129	0.206	**0.123**
Giants	16.20	**26.00**	24.86	24.95	0.265	**0.898**	0.844	0.839	0.274	**0.147**	0.270	0.299
Tools	16.19	**28.16**	27.54	26.73	0.575	**0.953**	0.938	0.896	0.250	**0.151**	0.204	0.167
Food	14.57	**23.68**	23.32	22.61	0.297	**0.832**	0.796	0.776	0.341	**0.203**	0.308	0.322
Pasta	12.50	**22.07**	21.23	20.64	0.216	**0.844**	0.789	0.715	0.271	**0.211**	0.311	0.283
Seasoning	17.31	**28.60**	27.79	27.12	0.412	**0.928**	0.898	0.881	0.279	**0.168**	0.276	0.263
Crest	15.91	**21.23**	20.30	20.11	0.209	**0.757**	0.670	0.653	0.304	**0.162**	0.315	0.410

Fig 9, we use 75%, 50%, and 25% of the original data for the experiment. Although the input images are dramatically decreased, our method still generates high-quality results, which retain the rainbow and background reflections, and the refraction details through the test tube. In Tab 4, note that even *with less input images for CD and Lab scenes*, our results are still comparable with or even better than NeRF and NeX with full number of inputs in the above challenging scenes.

4.5. Applications

As applications of NeuLF, we show results of depth estimation and automatic refocusing.

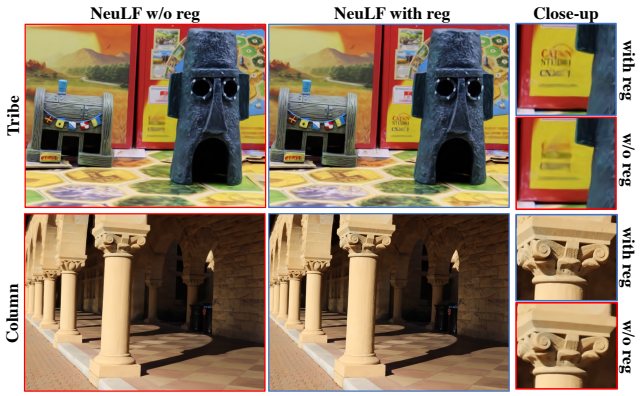

Figure 8: *Ablation study. We show results of NeuLF trained with and without multi-view and depth regularization on two Lambertian scenes. The last column shows the close-up comparison of the two scenes.*

Table 3: *A quantitative comparison of NeuLF trained with and without MVDL. The scores are computed cross test views of Tribe and Column scene.*

	PSNR↑		SSIM↑		LPIPS↓	
	w/o MVDL	w MVDL	w/o MVDL	w MVDL	w/o MVDL	w/ MVDL
Tribe	25.62	**26.93**	0.788	**0.830**	0.232	**0.219**
column	28.58	**29.20**	0.838	**0.847**	0.227	**0.211**

Table 4: *Impacts from the number of inputs. We input different numbers of images to evaluate the performance of our model.*

	method	#Images	PSNR ↑	SSIM ↑	LPIPS ↓
CD 75%	Ours	230	31.81	0.959	0.126
CD 50%	Ours	153	31.41	0.953	0.145
CD 25%	Ours	77	30.16	0.948	0.170
CD 100%	NeRF	307	30.14	0.937	0.206
CD 100%	NeX	307	31.43	0.958	0.129
Lab 75%	Ours	227	31.87	0.949	0.097
Lab 50%	Ours	151	31.74	0.948	0.104
Lab 25%	Ours	76	30.61	0.939	0.116
Lab 100%	NeRF	303	29.60	0.936	0.182
Lab 100%	NeX	303	30.43	0.949	0.146

In Fig 10, we show an example of the depth estimation and refocusing effect on our own captured scene Tribe. Note the detailed depth of the house and statue are successfully recovered. We can observe blurred depth around object boundary. Since our depth estimation is based on multi-view consistency, inaccuracy in camera pose estimation breaks the color consistency assumption around depth discontinuities and leads to depth errors.

With free-viewpoint scene depth, we can automatically select a focal plane given an image pixel location. We show two synthesized novel views rendered from our own captured scene. Then, we show the auto-refocus result given two locations on the image, one focuses on the near object (red "x"), and another on the far object

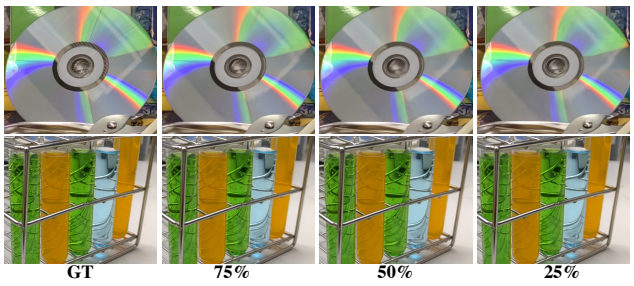

Figure 9: *Study on number of inputs. As shown in the figure, we input 75%, 50%, and 25% of the original images on the CD and Lab scenes.*

(green "x"). This is enabled by the dynamic 4D light field representation [IMG00].

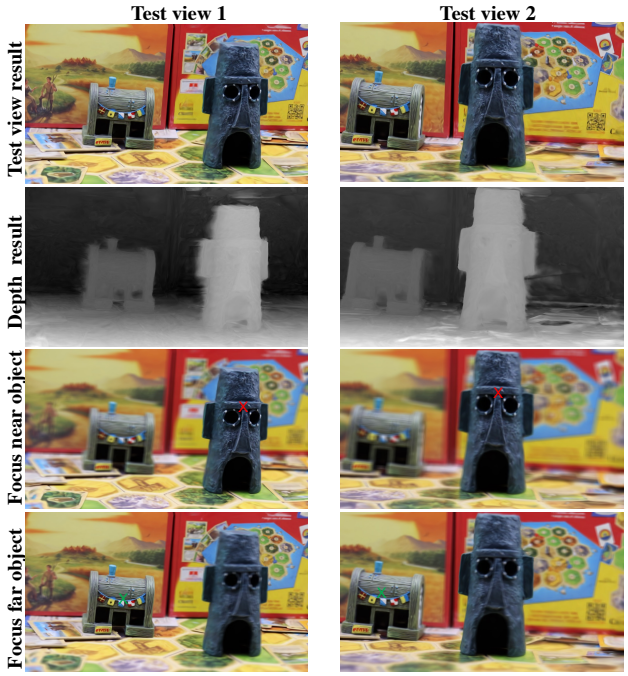

Figure 10: *Depth estimation and auto-refocus result. We show the results of two novel views (first row) and the depth estimation (second row). In the third row we refocus on the near object, and in the fourth row we refocus on the far object.*

4.6. Failure Cases

Our method is based on a 4D 2PP (two-plane parameterization) light field representation. Since each 3D world position corresponds to multiple discontinued 4D coordinates, NeuLF representation is difficult to learn. Therefore, we observe that our model cannot fully recover the high-frequency details in the scene as shown in Fig. 11. The seaweed texture (first row) and object's

hollow-out structure (second row) are over-smoothed. In addition, different exposure and lighting change cross the frames can lead to flickering artifacts in the results. Recent works show that using a high-dimensional embedding [TSM*20], or using periodic activation functions(position encoding) [SMB*20] can help recover fine details. However, we found that the above methods will cause over fitting on the training views and lead to works results on test views using NeuLF. Learning how to recover the fine details of the 2PP light field representation can be an interesting direction in the future.

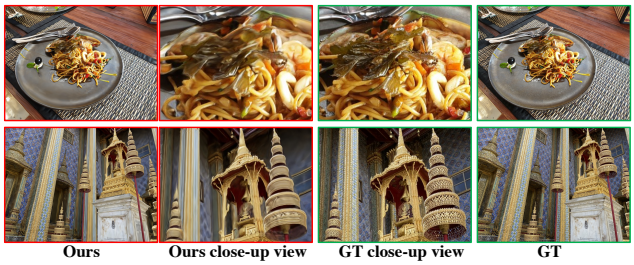

Figure 11: *Our failure cases. We show synthesized novel views and groundtruth. On the close-up views, we can see that our results lack fine details compared with ground truth.*

5. Conclusion

We propose a novel view synthesis framework called Neural 4D Light Field (NeuLF). Unlike NeRF, we represent a continuous scene using a 4D light field and train an MLP network to learn this mapping from input posed images. By limiting novel view synthesis to include only front views, NeuLF can achieve a comparable quality level as NeRF, but is hundreds times faster. speedup. Moreover, because the speedup is enabled by modeling the color of light rays, NeuLF does not need additional storage for acceleration. To optionally output per-ray depth, we propose two loss terms: multi-view consistency loss and depth loss. This enables synthetic auto-refocus effect. We demonstrate state-of-the-art novel view synthesis results, especially for scenes with reflection and refraction. We also demonstrate the effectiveness of our method with much fewer input images compared with NeRF and NeX.

6. Limitations and Future Work

There are several limitations to our approach. First, the novel viewpoints are limited to be on the one side of the two light slabs. In the future, we would like to extend the method to use more flexible 4D parameterizations such as multiple two planes, two cylindrical surfaces, or two spherical surfaces. By assuming the color is constant along a ray in free space, NeuLF cannot model rays that are blocked by the scene itself; therefore, novel viewpoints are always outside of the convex hull of the scene. This is an inherited limitation from light field.

Instead of using a 4D parameterization, lower-dimensional parameterization for specific applications can also be used. For example, in the work of concentric mosaic [SH99], by constraining camera motion to planar concentric circles, all input image rays are indexed in three parameters. By adopting this parameterization, a more compact representation of the scene can be achieved, which potentially runs even faster than a 4D parameterization.

Free-viewpoint video can be a straightforward extension of NeuLF from static scenes to dynamic ones. In the future, we would like to explore the possibility of including time in the formulation following [LSS*19].

Although our simplified NeuLF model can significantly improve the rendering speed compared with NeRF, it also has the limitations when it comes to 3D scene structure recovery. In the future, we would like to extend our work to reconstruct the surface from the reconstructed light field by using existing approaches such as Shape from Light Field (SfLF) techniques [HYP17].

References

[AHZ*22] ATTAL B., HUANG J.-B., ZOLLHÖFER M., KOPF J., KIM C.: Learning neural light fields with ray-space embedding networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022). 2

[BBM*01] BUEHLER C., BOSSE M., MCMILLAN L., GORTLER S., COHEN M.: Unstructured lumigraph rendering. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques* (2001), SIGGRAPH '01, p. 425–432. 1, 2

[BMT*21] BARRON J. T., MILDENHALL B., TANCIK M., HEDMAN P., MARTIN-BRUALLA R., SRINIVASAN P. P.: Mip-NeRF: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 5855–5864. 3

[CFG*15] CHANG A. X., FUNKHOUSER T., GUIBAS L., HANRAHAN P., HUANG Q., LI Z., SAVARESE S., SAVVA M., SONG S., SU H., ET AL.: ShapeNet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012* (2015). 4

[CRL20] CHEN B., RUAN L., LAM M.-L.: LFGAN: 4d light field synthesis from a single RGB image. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) 16*, 1 (2020), 1–20. 2

[FBD*19] FLYNN J., BROXTON M., DEBEVEC P., DUVALL M., FYFFE G., OVERBECK R. S., SNAVELY N., TUCKER R.: DeepView: High-quality view synthesis by learned gradient descent. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2019). 3

[FV21] FENG B. Y., VARSHNEY A.: SIGNET: Efficient neural representation for light fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 14224–14233. 2, 4

[GGSC96] GORTLER S. J., GRZESZCZUK R., SZELISKI R., COHEN M. F.: The lumigraph. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques* (1996), SIGGRAPH '96, p. 43–54. 1, 2, 4

[GKJ*21] GARBIN S. J., KOWALSKI M., JOHNSON M., SHOTTON J., VALENTIN J.: FastNeRF: High-fidelity neural rendering at 200fps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 14346–14355. 3

[HPP*18] HEDMAN P., PHILIP J., PRICE T., FRAHM J.-M., DRETTAKIS G., BROSTOW G.: Deep blending for free-viewpoint image-based rendering. *ACM Transactions on Graphics (TOG) 37*, 6 (2018), 1–15. 3

[HSM*21] HEDMAN P., SRINIVASAN P. P., MILDENHALL B., BARRON J. T., DEBEVEC P.: Baking neural radiance fields for real-time view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 5875–5884. 2

[HYP17] HEBER S., YU W., POCK T.: Neural epi-volume networks for shape from light field. In *Proceedings of the IEEE International Conference on Computer Vision* (2017), pp. 2252–2260. 9

[IMG00] ISAKSEN A., MCMILLAN L., GORTLER S. J.: Dynamically reparameterized light fields. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques* (2000), SIGGRAPH '00, p. 297–306. 2, 4, 5, 8

[KB14] KINGMA D. P., BA J.: Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014). 6

[KJJ*21] KELLNHOFER P., JEBE L. C., JONES A., SPICER R., PULLI K., WETZSTEIN G.: Neural lumigraph rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 4287–4297. 2, 3

[KTEM18] KANAZAWA A., TULSIANI S., EFROS A. A., MALIK J.: Learning category-specific mesh reconstruction from image collections. In *Proceedings of the European Conference on Computer Vision (ECCV)* (2018), pp. 371–386. 3

[LGL*20] LIU L., GU J., LIN K. Z., CHUA T.-S., THEOBALT C.: Neural sparse voxel fields. In *Advances in Neural Information Processing Systems* (2020), vol. 33. 3

[LH96] LEVOY M., HANRAHAN P.: Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques* (1996), pp. 31–42. 1, 2, 4, 6

[LMW21] LINDELL D. B., MARTEL J. N., WETZSTEIN G.: AutoInt: Automatic integration for fast neural volume rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 14556–14565. 3

[LSS*19] LOMBARDI S., SIMON T., SARAGIH J., SCHWARTZ G., LEHRMANN A., SHEIKH Y.: Neural volumes: Learning dynamic renderable volumes from images. *arXiv preprint arXiv:1906.07751* (2019). 9

[LSS*21] LOMBARDI S., SIMON T., SCHWARTZ G., ZOLLHOEFER M., SHEIKH Y., SARAGIH J.: Mixture of volumetric primitives for efficient neural rendering. *ACM Transactions on Graphics (TOG) 40*, 4 (2021), 1–13. 3

[MBRS*21] MARTIN-BRUALLA R., RADWAN N., SAJJADI M. S., BARRON J. T., DOSOVITSKIY A., DUCKWORTH D.: NeRF in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 7210–7219. 3

[MSOC*19] MILDENHALL B., SRINIVASAN P. P., ORTIZ-CAYON R., KALANTARI N. K., RAMAMOORTHI R., NG R., KAR A.: Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG) 38*, 4 (2019), 1–14. 3

[MST*20] MILDENHALL B., SRINIVASAN P. P., TANCIK M., BARRON J. T., RAMAMOORTHI R., NG R.: NeRF: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision* (2020), Springer, pp. 405–421. 2, 3, 6

[MWLL20] MENG N., WU X., LIU J., LAM E.: High-order residual network for light field super-resolution. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2020), vol. 34, pp. 11757–11764. 2

[NSP*21] NEFF T., STADLBAUER P., PARGER M., KURZ A., MUELLER J. H., CHAITANYA C. R. A., KAPLANYAN A., STEINBERGER M.: DONeRF: Towards real-time rendering of compact neural radiance fields using depth oracle networks. *Computer Graphics Forum 40*, 4 (2021), 45–59. 3

[PSB*21] PARK K., SINHA U., BARRON J. T., BOUAZIZ S., GOLDMAN D. B., SEITZ S. M., MARTIN-BRUALLA R.: Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 5865–5874. 3

[RK20] RIEGLER G., KOLTUN V.: Free view synthesis. In *European Conference on Computer Vision* (2020), Springer, pp. 623–640. 3

[RPLG21] REISER C., PENG S., LIAO Y., GEIGER A.: KiloNeRF: Speeding up neural radiance fields with thousands of tiny MLPs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 14335–14345. 2, 3

[SDZ*21] SRINIVASAN P. P., DENG B., ZHANG X., TANCIK M., MILDENHALL B., BARRON J. T.: NeRV: Neural reflectance and visibility fields for relighting and view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 7495–7504. 3

[SESM22] SUHAIL M., ESTEVES C., SIGAL L., MAKADIA A.: Light field neural rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 8269–8279. 2, 4

[SF16] SCHONBERGER J. L., FRAHM J.-M.: Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), pp. 4104–4113. 6

[SH99] SHUM H.-Y., HE L.-W.: Rendering with concentric mosaics. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (1999), pp. 299–306. 9

[SMB*20] SITZMANN V., MARTEL J., BERGMAN A., LINDELL D., WETZSTEIN G.: Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems 33* (2020). 9

[SRF*21] SITZMANN V., REZCHIKOV S., FREEMAN W. T., TENENBAUM J. B., DURAND F.: Light field networks: Neural scene representations with single-evaluation rendering. In *Advances in Neural Information Processing Systems* (2021). 2, 3

[STB*19] SRINIVASAN P. P., TUCKER R., BARRON J. T., RAMAMOORTHI R., NG R., SNAVELY N.: Pushing the boundaries of view extrapolation with multiplane images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), pp. 175–184. 3

[STH*19] SITZMANN V., THIES J., HEIDE F., NIESSNER M., WETZSTEIN G., ZOLLHÖFER M.: DeepVoxels: Learning persistent 3d feature embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019). 3

[SZW19] SITZMANN V., ZOLLHOEFER M., WETZSTEIN G.: Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *Advances in Neural Information Processing Systems* (2019), vol. 32. 3

[TFT*20] TEWARI A., FRIED O., THIES J., SITZMANN V., LOMBARDI S., SUNKAVALLI K., MARTIN-BRUALLA R., SIMON T., SARAGIH J., NIESSNER M., PANDEY R., FANELLO S., WETZSTEIN G., ZHU J.-Y., THEOBALT C., AGRAWALA M., SHECHTMAN E., GOLDMAN D. B., ZOLLHÖFER M.: State of the Art on Neural Rendering. *Computer Graphics Forum (EG STAR 2020)* (2020). 1, 3

[TSM*20] TANCIK M., SRINIVASAN P., MILDENHALL B., FRIDOVICH-KEIL S., RAGHAVAN N., SINGHAL U., RAMAMOORTHI R., BARRON J., NG R.: Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems 33* (2020), 7537–7547. 9

[WAA*00] WOOD D. N., AZUMA D. I., ALDINGER K., CURLESS B., DUCHAMP T., SALESIN D. H., STUETZLE W.: Surface light fields for 3d photography. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques* (2000), pp. 287–296. 2

[WLFC21] WU G., LIU Y., FANG L., CHAI T.: Revisiting light field rendering with deep anti-aliasing neural network. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021). 2

[WPYS21] WIZADWONGSA S., PHONGTHAWEE P., YENPHRAPHAI J., SUWAJANAKORN S.: NeX: Real-time view synthesis with neural basis expansion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021). 2, 3, 6

[WWX*21] WANG Z., WU S., XIE W., CHEN M., PRISACARIU V. A.: NeRF–: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064* (2021). 3

[YKM*20] YARIV L., KASTEN Y., MORAN D., GALUN M., ATZMON M., RONEN B., LIPMAN Y.: Multiview neural surface reconstruction by disentangling geometry and appearance. *Advances in Neural Information Processing Systems 33* (2020). 3

[YLT*21] YU A., LI R., TANCIK M., LI H., NG R., KANAZAWA A.: Plenoctrees for real-time rendering of neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 5752–5761. 2, 3

[ZFT*21] ZHANG X., FANELLO S., TSAI Y.-T., SUN T., XUE T., PANDEY R., ORTS-ESCOLANO S., DAVIDSON P., RHEMANN C., DEBEVEC P., ET AL.: Neural light transport for relighting and view synthesis. *ACM Transactions on Graphics (TOG) 40*, 1 (2021), 1–17. 3

[ZLY*17] ZHANG Y., LI Z., YANG W., YU P., LIN H., YU J.: The light field 3d scanner. In *2017 IEEE International Conference on Computational Photography (ICCP)* (2017), IEEE, pp. 1–9. 2

[ZRSK20] ZHANG K., RIEGLER G., SNAVELY N., KOLTUN V.: NeRF++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492* (2020). 3

[ZTF*18] ZHOU T., TUCKER R., FLYNN J., FYFFE G., SNAVELY N.: Stereo magnification: Learning view synthesis using multiplane images. *ACM Transactions on Graphics (TOG) 37*, 4 (July 2018). 2, 3