NeurMiPs: Neural Mixture of Planar Experts for View Synthesis

Zhi-Hao Lin[1,2] Wei-Chiu Ma[3] Hao-Yu Hsu[2] Yu-Chiang Frank Wang[2] Shenlong Wang[1]
[1]University of Illinois at Urbana-Champaign [2]National Taiwan University
[3]Massachusetts Institute of Technology

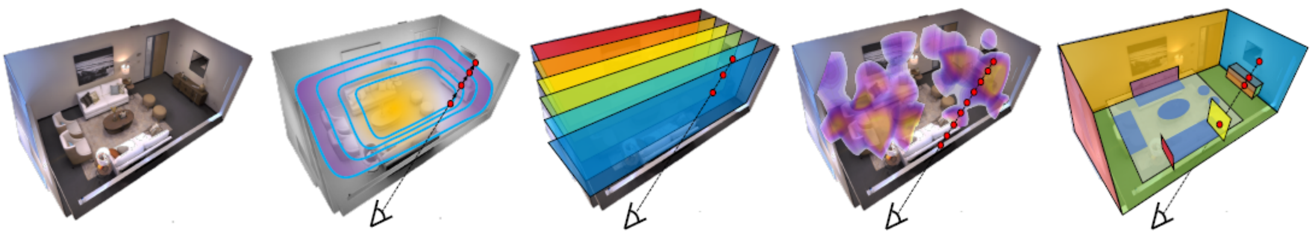


Figure 1. **Comparison between different 3D representations for neural rendering.** *Neural implicit surface* models accurate surface geometry but rendering requires expensive sequential sampling. *Multi-planar imagery* is efficient, but does not reflect true geometry and is not suitable for extrapolation. While NeRF is flexible, it is not sample-efficient and its learned density field might not reflect true scene geometry. Our proposed *Mixture of Planar Experts* is efficient and is able to model complicated surface geometry and appearance.

Abstract

We present Neural Mixtures of Planar Experts (NeurMiPs), a novel planar-based scene representation for modeling geometry and appearance. NeurMiPs leverages a collection of local planar experts in 3D space as the scene representation. Each planar expert consists of the parameters of the local rectangular shape representing geometry and a neural radiance field modeling the color and opacity. We render novel views by calculating ray-plane intersections and composite output colors and densities at intersected points to the image. NeurMiPs blends the efficiency of explicit mesh rendering and flexibility of the neural radiance field. Experiments demonstrate superior performance and speed of our proposed method, compared to other 3D representations in novel view synthesis.

1. Introduction

Metaverse is coming. Imagine one day in the future. People can explore the world freely and immersively without leaving their room. When they move forward, details pop up; when they move sideways, the occluded regions re-appear. Whenever people take action, the metaverse will respond with corresponding visual scenes that look natural, just as if people are visiting the place in person. While appealing, bringing this vision to reality requires advancement in multiple domains, one of which is real-time, high-quality, memory-efficient novel view synthesis. Specifically, given a set of posed images of the world, an ideal NVS system has to be able to photo-realistically re-render the scene from novel viewpoints such that people cannot tell the difference. The system also needs to be fast and lightweight such that it can be deployed ubiquitously.

Towards this grand goal, researchers have developed a plethora of methods to reproduce our visual world. One promising direction is to explicitly model the geometry of the scene (*e.g.* multi-planar imagery [12, 13, 73, 85], point clouds [1, 39, 53], meshes [31, 51, 52]) and conduct image-based rendering (IBR) [3, 9, 10, 31, 63]. By adapting visual features from other existing views, these approaches can render high-quality images efficiently. Unfortunately, they are often memory intensive and require good proxy geometry. On the other hand, recent advances in neural radiance fields [44, 50, 78, 79, 82, 84] have allowed us to synthesize highly realistic images with low memory footprint. By encoding color and density functions as neural networks, they can handle complicated geometry and scene effects that are difficult for conventional methods, *e.g.*, thin structures, specular reflections, and semi-transparent objects. The flexibility of volume rendering, however, is a double-bladed sword. Without proper surface modeling, they cannot capture the scene geometry accurately, resulting in artifacts during view-extrapolation setup.

With these motivations in mind, we aim to find an alternative 3D scene representation that is compact, efficient, expressive, and generalizable. Specifically, we investigate planes, one of the simplest geometric primitives yet powerful for representing complicated scenes. Most surfaces

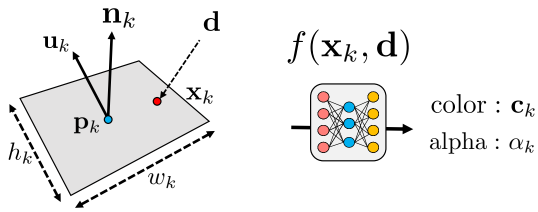
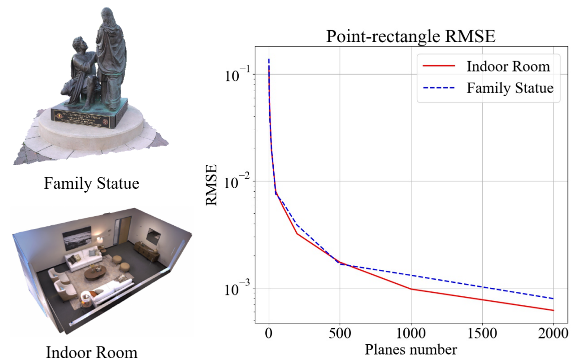


Figure 2. **Planar Expert Parameterization** Left: each plane consists of a 3D center, a plane normal, an up vector, and width and height; Right: the appearance is modeled through a neural radiance field function, which takes the 3D coordinate and ray direction as input and outputs color and opacity.



(a) Target Scenes (b) # of Planes vs Point-to-Plane Error

Figure 3. **Surface Fitting using a Mixture of Planar Sprites** Left: target scenes; Right: fitting results.

in man-made environments are locally planar. Taking the scene in Fig. 3 as an example, one could use 500 planes to fit a 5×5 m^2 scene with a maximum point-to-surface error of 8.66 mm. This result indicates that we might consider modeling our real-world surfaces through piece-wise local planar structures. The local planar world is not a surprisingly new concept to many researchers in vision [35, 41, 48]. It also profoundly impacts the graphics community and is the most common representation for rendering.

Unlike multi-planar imagery [12, 68, 73, 85], which represents the scene through frontal-parallel planes, our proposed approach allows each plane to have an arbitrary position, direction, and size. Consequently, our representation is more flexible to approximate the scene geometry. Unlike volume rendering, NeurMiPs explicitly models surface using planar geometry. Hence fast rendering can be done through the efficient ray-plane intersection and eliminate the computations in empty spaces. Fig. 1 depicts our proposed 3D representation and a comparison to other representations for neural rendering.

We validate our approach on several standard benchmarks for novel-view synthesis. The experiments demonstrate that our end-to-end method is significantly faster than the volume-based method with similar or better rendering quality and performs favorably against surface-based neural rendering methods with higher rendering quality and memory reduction. Furthermore, we evaluate our approach on a new challenging benchmark for view extrapolation, demonstrating superior performance compared to other state-of-the-art methods. In particular, NeurMiPs outperforms NeRF with over 1dB PSNR gain at significant novel testing views. The explicit planar surface representation of NeurMiPs could also readily be employed in modern graphics engines.

2. Related Work

Our approach is closely related to classical work on image-based modeling and rendering, as well as recent learning-based efforts. We also draw inspiration from the prior art on planar scene representations. In this section, we briefly review previous work in these two major directions.

2.1. Novel View Synthesis

Explicit surface modeling: Pioneering view synthesis works leverage explicit surface geometry. Mesh representations have been adopted as proxy geometry to guide the image-based warping, from the source views to a target pose [7,9,10,31]. Dense point clouds and surfels are alternate explicit representations of polygonal meshes [1, 39, 53, 75]. Both are suitable for hardware acceleration. Hence, they have superior efficiency and high rendering quality for synthetic data. It is, however, challenging to handle imperfect geometry and view-dependent effects. Recent works investigate learning approaches for explicit surface rendering [51, 52, 66, 67] or using 2D network to retouch the images [39, 72]. NeurMiPs is a form of explicit geometry. Thus we inherit its speed advantage. Unlike most explicit geometry methods, we leverage the neural radiance field to improve rendering quality while being memory efficient.

Multi-plane imagery: Another closely related structure is the layered imagery, including multi-plane imagery (MPI) and layered depth imagery (LDI) [3, 12, 13, 24, 43, 58, 60, 73, 85]. MPI represents the scene using a stack of frontal-parallel images. It allows fast rendering and can deliver photorealistic results with slight and frontal-parallel movement. However, its layered geometry structure brings artifacts 360° surrounding views or sagittal plane movement (*e.g.* walking or flying). Recent works extend the view extrapolation ability [43] by fusing multiple MPIs with additional memory cost. One of the most related MPI work to us is NeX [73]. Both utilize multi-planar geometry and neural radiance function. However, NeX uses fixed, frontal-parallel planes. In contrast, ours uses learnable, slanted planes, bringing more flexibility to handle complicated scenes and render extrapolated and surrounding views.

Implicit surface: The limitations of explicit geometry could be alleviated through implicit surface modeling [46]. Recent works start to jointly model surface and appearance using neural representation [74, 76, 77, 81]. They achieve

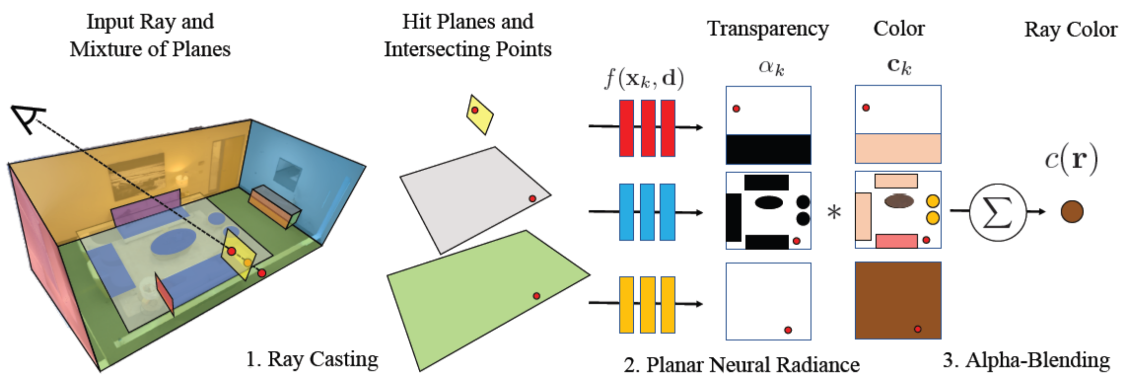


Figure 4. **The Rendering Pipeline of NeurMiPs.** We first cast rays and identify intersecting points and planes. Color and opacity can then be evaluated through each plane's neural radiance field. Finally, alpha-blending step is conducted to output the final ray color.

state-of-the-art reconstruction quality and decent view synthesis results. However, rendering the implicit function requires sequential ray marching steps, and additional steps are required to extract the surface.

Neural volumetric rendering: Volumetric radiance field can be dated back to the late 90s [19]. Recent works, such as NeRF [44] and Neural Volumes [37] start to investigate deep learning for volume rendering. It leverages the expressiveness of neural net and the flexibility of volume rendering. A plethora of new approaches have been proposed during the past year to extend NeRF [4, 17, 25, 36, 42, 44, 50, 71, 82, 84]. Representative works can handle sparse input views [69, 79], unbounded scenes [82], overcome aliasing effect [4] and take as input unknown/noisy poses [33].

The seminal NeRF [44] does not render in real-time. Several works attempt to accelerate NeRF using different strategies. For example, methods like [17, 49, 50, 78] choose to decompose the input scene into smaller regions and use smaller networks to model 3D geometry for each. Other approaches reduce the amount of samples per ray, through early ray termination [47], empty space skipping [50, 78], learnable sparse sampling [2, 45, 64], or closed-form sampling-free integration [34]. Deferred rendering or baking techniques have also been used to accelerate NeRF [17, 25]. Our approach is a new instantiation of the above acceleration techniques through the planar representations. Perhaps closet to our work is MVP [38] who also take advantage of geometric primitives. There, however, exists a few key differences. First, while MVP exploits *dense voxel grids* to capture the sophisticated texture of human head, we model the scene structures with *planes*. Second, MVP explicitly generates RGBα for each voxel which is memory-consuming, whereas NeurMiPs models texture with neural nets.

2.2. Planar Scene Representation

We are not the first to realize the potential of multiple slanted planes for representing the scene geometry. The computer vision and graphics community has a long history of leveraging planar surfaces for modeling and rendering. Various forms of planar scene representations have been investigated [3, 14, 21, 23, 27, 28, 32, 55, 63]. Representative works include polygon mesh [5, 20], Marr sketch [40], Manhattan world [15, 26], Binary space partitioning tree [8], 3D box layout [22, 23], origami theory [28], slanted planes [3, 6, 63], *etc*. A closely related line of research is called Layered sprites [63], which shares a similar geometric representation to ours. The key difference lies in the appearance representation and rendering: layered sprites use image textures for each plane and rendering is done through homography warping, whereas our NeurMiPs exploits expressive neural radiance field, and rendering is done through ray casting, which captures better view-dependent effects and runs faster in complicated scenes.

Numerous methods have been developed to reason planar structures from images. For example, one can detect planes from images [35, 80], recover meshes [18], reconstruct slanted planar surfaces from stereo [6, 16], leverage planar structure for SLAM [54], estimate surface normal and boundary based on the local planar assumption [14] and finally reconstruct planar spirits from multiple images [27, 55]. The proposed NeurMiPs can be treated as a multi-view slanted plane reconstruction method by minimizing the photo-metric rendering loss.

3. Method

In this work, we tackle the problem of novel view synthesis. Our goal is to improve the rendering efficiency as much as possible while improving the rendering quality at extreme novel views. Towards this goal, we propose a novel neural representation called the mixture of planer experts and design a neural rendering method using NeurMiPs.

Specifically, we first represent the scene as a mixture of local planar surfaces. Every local surface is an oriented 2D rectangle in 3D. We then use a neural radiance field function for each plane to encode its view-dependent appearance and transparency. Both the geometry and the radiance fields are learned end-to-end from the input images. During the

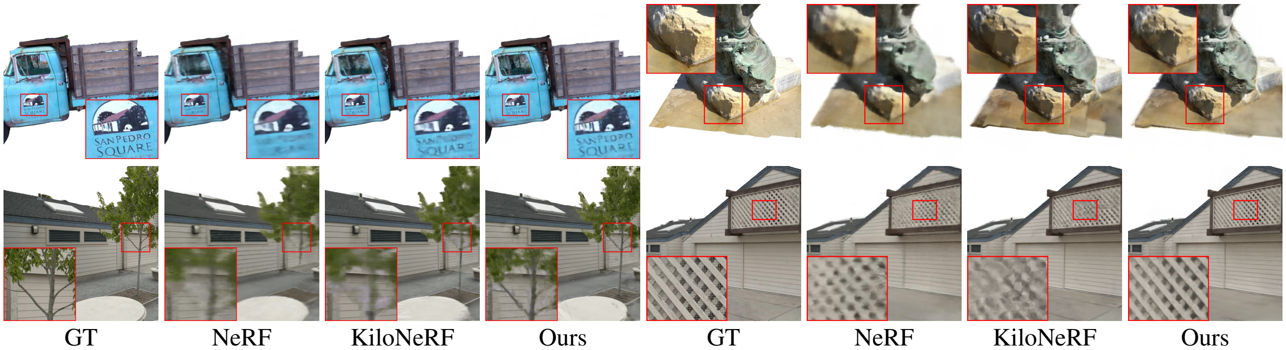


Figure 5. **Qualitative Results of Tanks & Temples.** Zoom in for better visual comparisons.

rendering time, our method will first conduct a ray-rectangle intersection check. Each ray will only hit a small subset of surfaces. The color and transparency will be evaluated based on intersecting point's coordinate. Finally, the ray color will be calculated through alpha blending the colors of all intersecting points.

Fig. 1 compares different 3D representations for view synthesis. Compared to neural surface rendering, our method is efficient in both memory and computation; the approach is significantly more sample-efficient than volume rendering with better extrapolation; compared to multi-plane imagery, our approach better reflects the geometry.

3.1. Mixture of Planar Experts

The mixture of planar experts representation consists of K rectangular surface parameterized by $\{\mathbf{s}_k = (\mathbf{p}_k, \mathbf{n}_k, \mathbf{u}_k, w_k, h_k)\}$, where $\mathbf{p}_k$ is the rectangle center; $\mathbf{n}_k$ is a normalized 3D vector representing the plane's normal; $\mathbf{u}_k$ is the normalized up vector defines the y-axis direction of the plane coordinate in the world; (w_k, h_k) is the plane's size. Inspired by the success of recent neural rendering [44, 73], we represent each planar expert's view-dependent appearance and transparency as a 3D neural radiance field function

$$(\mathbf{c}_k, \alpha_k) = f_k(\mathbf{x}_k, \mathbf{d}) \tag{1}$$

The function takes the 3D space coordinate $\mathbf{x}_k$ and normalized 3D ray direction $\mathbf{d} = (d_x, d_y, d_z) \in \mathbb{S}^2$ as input and output the corresponding color and transparency. See Fig. 2 for an illustration of the planar representation.

Network Architecture Each planar sprite only needs to model a local slice of the full radiance field. Hence, we use a significantly smaller multi-layer perceptron (MLPs) for each planar expert model. Each MLP model consists of three fully-connected hidden layers, with ReLU activation for each hidden layer and sigmoid activation for final output. The networks predict both color and alpha values. Following recent works [44, 65], the ray input is transformed into a higher dimensional space with high-frequency functions before passing to the network, which enhances the capability of capturing high-frequency textures.

Representing the Scene Geometry A natural question arises: is it sufficient for representing the complicated world with a mixture of planes? To answer this question, we conduct a quick experiment to demonstrate its power. Specifically, we choose two complicated 3D scenes, an indoor environment from Replica dataset [61] and an outdoor scene from Tanks and Temple [30]. Both scenes consist of sufficiently complicated geometric structures like trees, poles, circular shape surfaces. We use a mixture of planar experts to fit the surface geometry, by minimizing the point to plane distance for points sampled from the scene's surface. Fig. 3 illustrates the local planar surface fitting performance as a curve of the number of rectangles vs. average point-to-plane distance. From the figure, we could see that with only 1000 planes, we could reach 10^{-3} RMSE point-to-plane error, with the whole scene normalized into a unit sphere. These results suggest the multi-plane surface geometry is suitable to represent complicated scenes for neural rendering.

3.2. Rendering

At the testing time, a pixel is rendered by shooting a ray from the eye and evaluating the radiance along the ray. Formally, the input ray consists of a original point and a normalized directional vector $\mathbf{r} = \{\mathbf{o}, \mathbf{d}\}$ where $\mathbf{o}$ is the origin and $\mathbf{d}$ is the direction.

Ray-Plane Intersection Rendering NeurMiPs is very efficient thanks to its simple geometry. The first step is the ray-rectangle intersection: decide whether a local plane is intersecting the ray. This intersection check can be done analytically. Firstly, we will find the intersection between a given ray $\mathbf{r} = \{\mathbf{o}, \mathbf{d}\}$ and an infinite size plane $\{\mathbf{n}_k, \mathbf{p}_k\}$:

$$\mathbf{x}_k = \mathbf{o} + \frac{(\mathbf{p}_k - \mathbf{o}) \cdot \mathbf{n}_k}{\mathbf{d} \cdot \mathbf{n}_k} \mathbf{d} \tag{2}$$

We will only keep the intersected rectangles for the next phase. In practice, only a small fraction will be kept.

Radiance Evaluation We then evaluate the ray's color and transparency at the intersecting planar experts. Specifically, given the input coordinate transform $\mathbf{x}_{p,k}$ and the direction

Figure 6. **Qualitative Results of the Replica Dataset.**

$\mathbf{d}$ for each planar expert, we will output a transparency value α_k and a color $\mathbf{c}_k$ by evaluating its neural radiance function, according to Eq. 1.

Alpha Composition Now, each ray has collected a set of of point samples $\{\mathbf{x}_j, \mathbf{c}_j, \alpha_j\}$. We sort the point from closest to furthest to the eye $\mathbf{o}$, and conduct alpha composition to get the final estimation of the ray's color:

$$c(\mathbf{r}) = \sum_j \prod_i^{j-1} (1 - \alpha_i)\alpha_j \mathbf{c}_j \tag{3}$$

Fig. 4 depicts the detailed procedure of our rendering process per each ray. We want to highlight three important properties of our method. First, thanks to our mixture-of-planar structure, the ray-geometry intersection is computed efficiently in closed-form, which is extremely efficient. Second, each ray will only hit a handful of planes. This results in a small number of samples we need to evaluate for each ray, significantly boosting the speed. Finally, each planar expert radiance function only needs to model a local surface. Hence, the required computation for the radiance MLPs is also significantly smaller than NeRF.

3.3. Training

Training NeurMiPs requires jointly optimizing the plane geometry $\{(\mathbf{c}_k, \mathbf{n}_k, \mathbf{u}_k, w_k, h_k)\}$ and the radiance $f_k(\cdot, \cdot)$. Training from scratch results in artifacts and low-fidelity geometric structures. In practice we observed better results can be acquired through two training techniques: 1) planar initialization through geometric loss minimization; 2) distillation from a large teacher radiance model.

Plane Initialization We initialize the plane geometric parameters $\{\mathbf{s}_k\}$ using the coarse 3D point cloud $\{\mathbf{x}_i\}$ estimated by structure-from-motion [56]. Specifically, we optimize the following point-to-rectangle distance function:

$$\mathcal{L}_g = \sum_i \min_k d(\mathbf{x}_i, \mathbf{s}_k) + \lambda \sum_k (w_k h_k)^2, \tag{4}$$

where $\min_k d(\mathbf{x}, \mathbf{s_k})$ is the distance from point x to the closest rectangular surface s_k. An area regularization $(w_k h_k)^2$ is adopted to forbid the rectangle to be arbitrarily large.

Distillation After the planar geometry initialization, we then jointly optimize radiance and geometry. Inspired by the success of NeRF distillation [50, 62, 78], we first train a large-capacity, ordinary NeRF as the teacher model to distill knowledge from. It follows standard NeRF neural network architecture with two noticeable differences. First, the point sampling is conducted through NeurMiPs's ray-planar intersection. Second we also jointly optimize the planar experts' geometric parameters. The joint geometry and photo-metric loss are used for training the planar-guided NeRF:

$$\mathcal{L}_{total} = \mathcal{L}_g + \mathcal{L}_c,$$

where $\mathcal{L}_g$ is the point-to-rectangle geometry loss defined in Eq. 4 and $\mathcal{L}_c$ is the L2-photometric loss

$$\mathcal{L}_c = \sum_{\mathbf{r}} \|c(\mathbf{r}) - c_{\text{gt}}(\mathbf{r})\|_2^2. \tag{5}$$

For each planar expert, we then distill knowledge from the teacher model by minimizing the difference between the teacher's output and each student's output. Specifically, we draw random points uniformly from the rectangle and random view directions from the half unit sphere for each batch. Student network's parameters are updated by minimizing the L2 loss of both alpha values and the colors.

Fine-Tuning After over-fitting to the teacher network, we will fix plane parameters and fine-tune our student radiance field models to further improve the rendering quality. Specifically, we minimize the L2-photometric loss $\mathcal{L}_c$ between the rendered pixel color and the ground-truth color.

3.4. Implementation

Despite being efficient by design, NeurMiPs will benefit from several techniques during implementation to further boost the rendering speed, making it to be real-time.

Alpha Baking Another acceleration technique is texture pre-baking. Inspired by prior work on pre-caching NeRF [17, 73, 78], we propose to pre-render alpha values and bake them as alpha textures for each rectangular plane. To be more specific, the view-independent alpha is baked into each plane i as texture maps A_i of size $w \times h$. During inference, when a ray hits a surface, we could retrieve its corresponding

Model	PSNR↑	SSIM ↑	LPIPS ↓
NeRF [44]	28.32	0.904	0.168
SRN [59]	24.10	0.847	0.251
Neural Volumes [37]	23.70	0.834	0.260
NSVF [36]	28.40	0.900	0.153
PlenOctrees [78]	27.99	**0.917**	0.131
KiloNeRF [50]	28.41	0.910	0.090
Ours	**28.46**	0.908	**0.089**

Table 1. **Quantitative comparison on Tanks & Temples.**

Model	PSNR↑	SSIM↑	LPIPS↓
NeX [73]	24.76	0.832	0.152
NeRF [44]	30.12	0.901	0.097
PlenOctrees* [78]	27.72	0.872	0.174
KiloNeRF* [50]	29.37	**0.904**	0.097
Ours	**30.80**	0.900	**0.088**

Table 2. **Quantitative comparison on Replica.** All models are evaluated on a single TITAN RTX. Please see text for more details.

alpha value by bilinear interpolating from the baked alpha texture A_i. Note that applying alpha baking directly during inference might bring a minor rendering performance drop. Therefore, we fine-tune the RGB branch of planar experts for better rendering quality.

Early Ray Termination Evaluating radiance for all intersecting rectangles is unnecessary, in particular when the transmittance value $\prod_i(1-\alpha_i)$ is close to zero (*e.g.* ray hitting a non-transparent plane) since it may only have a minor impact. In practice, we exploit early ray termination to avoid additional network evaluation, thereby enhancing rendering efficiency considerably. We also perform re-normalization if the sum of the alpha values is smaller than 1. We empirically observe that it will improve performance.

Custom CUDA kernel To further accelerate model inference, we implement a custom CUDA kernel for ray-plane intersection, model inference, and alpha composition. We fuse the network evaluation of each expert into a single CUDA kernel so that all experts can render in parallel. As we will show in Sec. and supp. materials, this improves rendering efficiency significantly.

4. Experiments

4.1. Experimental Setup

Datasets: We evaluate NeurMiPs on two challenging datasets: Tanks & Temple [30] and Replica [61]. Tanks & Temples consists of five bounded *real-world* scenes [36]. Each scene contains $152 \sim 384$ high-resolution images (1920×1080) captured from surrounding 360° viewpoints.

Replica is a *synthetic* dataset featuring a diverse set of indoor scenes. Each scene is equipped with high-quality geometry and photorealistic textures, allowing one to render high-fidelity images from arbitrary camera poses. The flexibility also allows us to generate challenging novel view synthesis scenarios that are not available in existing benchmarks (*e.g.*, extreme view extrapolation). In this work, we randomly select seven scenes and render 50 training images and 100 test images for each of them. The camera poses are sampled randomly within a pre-defined range. We adopt a wider range for the test split so that the test images can cover broader views and may include unseen areas. The setup allows to evaluate the extrapolation capability of existing approaches. We adopt BlenderProc [11] as our physical-based rendering engine. The image size is set to 512×512.

Metrics: Following [44,50], we adopt peak signal-to-noise ratio (PSNR), structural similarity index (SSIM) [70], and perceptual metric (LPIPS) [83] for quantitative evaluation.

Baselines: We compare our approach against state-of-the-art neural radiance field (*i.e.*, NeRF [44]), MPI-based method (*i.e.*, NeX [73]), and hybrid, real-time methods (*i.e.*, NVSF [36], KiloNeRF [50], PlenOctrees [78]). We refer the readers to supp. materials for more details.

Implementation details: For each scene, we initialize our plane geometry with the sparse point clouds from COLMAP [56, 57]. Plane centers $\{p_k\}$ are selected by farthest point sampling, and plane orientations $\{n_k, u_k\}$ are initialized as normals estimated from local point sets around p_k. We set the number of planar experts to be 500 for Replica and 1000 for Tanks and Temple. We first train the teacher model for 6K epochs, then we distill the planar experts for 1.5K epochs. Finally, we fine-tune the experts for 2.5K epochs. We use Adam [29] optimizer with a learning rate of 5×10^{-4} across all experiments.

4.2. Tanks & Temples

As shown in Tab. 1, our approach is comparable to or better than prior art on all three metrics. Specifically, while NeRF generates blurry textures for large-scale scenes and KiloNeRF produces blocking artifacts, NeurMiPs is able to capture detailed textures on planar surfaces with sharp boundaries. Furthermore, our local planar structure can handle non-planar and thin objects well through a combination of planar experts and alpha composition. See the tree branches and leaves in the *Barn* example in Fig. 5.

4.3. Replica

We further evaluate our approach on Replica. Since Replica consists of extrapolated views (as mentioned in Sec. 4.1) that have not been observed during training, previous voxel-based implicit methods such as PlenOctrees [78], KiloNeRF [50] suffer drastically. Those methods prune out redundant voxels during training. Therefore, they cannot

	Easy			Medium			Hard		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
NeX [73]	25.88	0.844	0.136	24.56	0.828	0.156	23.86	0.826	0.164
NeRF [44]	31.41	**0.915**	0.081	29.98	**0.901**	0.097	29.02	0.887	0.113
PlenOctree [78]	28.05	0.877	0.171	27.613	0.872	0.174	27.51	0.866	0.179
KiloNeRF [50]	28.00	0.908	0.085	28.06	0.893	0.105	27.50	0.886	0.117
Ours	**31.98**	0.914	**0.074**	**30.31**	0.894	**0.092**	**30.05**	**0.892**	**0.096**

Table 3. **Performance vs viewpoint difference.**

Model	Median ↓	Inlier (%) < 0.05	Inlier (%) < 0.10	Inlier (%) < 0.50
NeX [73]	0.767	3.3	6.6	33.3
NeRF [44]	0.137	19.6	38.3	**87.8**
KiloNeRF [50]	0.125	32.9	45.9	77.2
Ours	**0.066**	**43.7**	**59.0**	84.2

Table 4. **Peformance of estimated depth.**

SfM geometry	Distillation	PSNR	SSIM	LPIPS
		25.069	0.818	0.158
✓		30.810	0.909	0.082
✓	✓	33.659	0.941	0.051

Table 5. **Ablation study.** Scene: Replica kitchen. "SfM geometry" refers to planes initialization with point cloud extracted by COLMAP [56, 57].

estimate the appearances of unseen regions. To (partially) alleviate this issue, we reduce the pruning threshold so that voxels are preserved even if they have lower volume density, at the cost of larger memory footprint/slower inference speed. In contrast, NeurMiPs represents the scene geometry with multiple planes, and can generalize better in view extrapolation. As shown in Tab. 2 and Tab. 7, our approach reaches the best quality-efficiency trade-off and is comparable to or better than prior art on all three metrics. We refer the readers to the supp. material for implementation details of the baselines and more comprehensive comparison. We also show some qualitative results in Fig. 6. NeRF produces fog-like artifacts in free space, while NeX [73] has significant "stack-of-cards" effects. NeurMiPs, in contrast, has significantly better visual quality, even at extrapolated novel views.

4.4. Analysis

Performance vs viewpoint difference: To gain insights into when NeurMiPs performs the best, we divide the test split of Replica into three categories, namely easy, medium, and hard, based on the proximity to the nearest training views, and evaluate our model. As shown in Tab. 3, NeurMiPs is comparable to or better than competing methods across all settings. In particular, NeurMiPs improves the performance the most when the viewpoint difference is large (*i.e.*, 1.03 dB PSNR gain and 0.017 LPIPS score reduction).

Depth estimation: To verify how well NeurMiPs models the scene geometricy, we follow previous work [44] to generate depth map at each viewpoint and compare with those from the baselines. Specifically, we estimate the expected depth value $d(\mathbf{r})$ along each ray $\mathbf{r}$ by alpha composition:

$$d(\mathbf{r}) = \sum_{j} \prod_{i}^{j-1} (1-\alpha_i)\alpha_j t_j \tag{6}$$

where t_j is the depth of sampled points j. As shown in Tab. 4, our planar experts are flexible and are able to approximate scene geometry well in most cases. However, since the size of the plane is finite, the planes may not cover all regions in extreme viewpoints. The scenes that are modeled by the background thus induce higher depth error. Note that this can be resolved by adopting multiple-layer boxes. We leave this for future study.

Speed-memory trade-off: NeurMiPs models the scene with a mixture of planar experts. The compact representation of the planes and the associated tiny MLPs not only induces a significantly lower memory footprint compared to voxel-based approaches, the planar parameterization also allows us to exploit ray-plane intersection to sample query points efficiently for low-cost radiance evaluation. Together with our customized CUDA kernel, we can achieve 19.16 frames per second on Replica. Comparing to the baselines (see Tab. 7), NeurMiPs achieves the best speed-memory trade-off.

Training strategy: To validate the contribution of each training technique (Sec. 3.3), we evaluate our model with different combinations. As shown in Tab. 5, initializing plane geometry with sparse point clouds significantly improves the performance. We conjecture this is because good initialization allows the model to alleviate the shape-radiance ambiguity [82] and converge to the correct geometry. With the help of distillation, one can further reduce the artifact and improve the results. We hypothesize this is because the guidance of teacher model prevents our model from getting stuck at local minima. Both observations concur with the findings of previous works [50, 71]. We also note that one needs to conduct SfM to obtain the camera poses in practice, hence the sparse point cloud from SfM is essentially "free".

Performance w.r.t. number of planar experts: Since we aim to model the scene appearance and geometry with planar experts, one natural question to ask is: how well does the approach scale with the number of planar experts. As shown in Tab. 6, more planes in general leads to better results. This is reasonable as we can fit the scene much better. However, it may also increase the model size and reduce the efficiency due to more ray-plane intersections.

Specular effect: Similar to NeRF, our planar experts model view-dependent effects by taking viewing direction

# Planes	25	50	100	200	500	1000
# Params(M)	0.15	0.31	0.62	1.24	3.11	6.21
PSNR	26.41	27.69	29.19	30.10	30.87	30.64
SSIM	0.828	0.851	0.877	0.888	0.900	0.902
LPIPS	0.168	0.140	0.112	0.098	0.088	0.085

Table 6. **Effect of plane number.** Dataset: Replica. Note that #planes=500 achieves the best complexity-quality trade-off.

Ground-Truth Ours Rendering Geometry

Figure 7. **Visualizing the geometry of learned planes.** Each color is a plane learned by NeurMiPs.

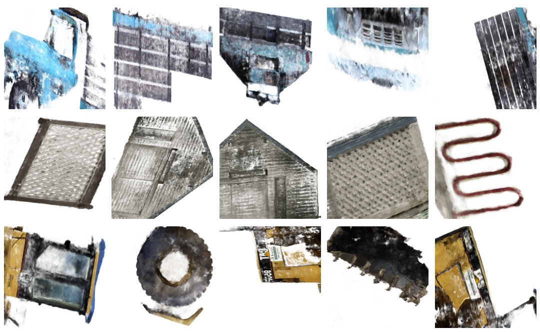


Figure 8. **Selected planar experts on Tanks & Temple**. We prebake the alpha and color values into a 2D texture for each planar expert, capturing diverse local surfaces with diverse appearance and geometry, e.g., the bike rack and the wheel.

d into account (see Eq. 1). We further alpha-composite radiances from all intersecting planes ($\sim$10 per ray) to compensate for the specular effect (similar to MPI). An example of specular windows is shown in Fig. 9.

Planar experts visualization: To better understand what is learned in the NeurMiPs, we visualize its planar spirits geometry, plane index, and textures. Specifically, we show the rendered surface colored by alpha-composed planar surface indices in Fig. 7. NeurMiPs learns to capture these structures with few large planes (denoted in the same color), while approximating the non-planar regions (*e.g.* tires, front of car) with more planes. Fig. 8 depicts a collection of learned textures maps baked from the radiance field of several planar experts. We see that comprehensive per-plane textures have been learned with sufficient interpretability.

Combining with graphics engine: One appealing property of our approach is its compatibility with polygon-mesh-based rendering engines. In fact, our representation could be considered as a polygon mesh with K rectangular faces. We can thus pre-bake ray colors into high-resolution, view-dependent textures for each plane and save the textured mesh representation of the scene. We can then write our view-dependent shader in OpenGL and render the scene using the standard rasterization engine. Note that depth sorting for each planar surface and back-to-front rendering is necessary to ensure the correct alpha blending procedure in Eq. 3. Texture baking brings notoriously acceleration at the cost of additional memory consumption and small rendering quality drop, due to the discretization of the continuous radiance field. The final accelerated rendering achieves 976 frames per second for the 1000-plane Truck scene at 1920x1080 resolution on a single RTX 3090 desktop.

	NeX	NeRF	PlenOctree*	KiloNeRF*	Ours
# Params (M)	21.28	1.19	1457.2	6.21	3.11
FPS	0.142	0.106	78.04	4.19	19.16

Table 7. **Model size and inference speed on Replica.**

Figure 9. **Specular effects.** Dataset: Tanks&Temples.

Limitations: Our method has several key limitations. First of all, NeurMiPs relies heavily on SfM point clouds for plane initialization (see Tab. 5). If the sparse point clouds are noisy or unavailable, our performance will degrade. Furthermore, our model currently cannot handle unbounded scenes. One possible solution is to incorporate techniques from NeRF++ [82] to model the background texture with non-euclidean coordinates. We leave this for future study.

5. Conclusion

In this paper, we proposed NeurMiPs, a novel 3D representation for novel view synthesis. NeurMiPs represents the 3D scene with a mixture of learnable planar experts. Each plane consists of a rectangular shape and a neural radiance field. Our approach alleviates the frontal-parallel limitation of MPI-based methods while remaining efficient thanks to the efficient ray-casting-based rendering. We demonstrated that our approach could be integrated with classic rendering pipelines. We believe NeurMiPs will open new possibilities for 3D modeling and rendering.

Acknowledgement We thank National Center for High-performance Computing (NCHC) for providing computational and storage resources. We also thank NVIDIA for hardware donation.

References

[1] Kara-Ali Aliev, Artem Sevastopolsky, Maria Kolos, Dmitry Ulyanov, and Victor Lempitsky. Neural point-based graphics. In *ECCV*. Springer, 2020. 1, 2

[2] Relja Arandjelović and Andrew Zisserman. Nerf in detail: Learning to sample for view synthesis. *in ArXiv*, 2021. 3

[3] Simon Baker, Richard Szeliski, and P Anandan. A layered approach to stereo reconstruction. In *CVPR*. IEEE, 1998. 1, 2, 3

[4] 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. *ICCV*, 2021. 3

[5] Bruce G Baumgart. Winged edge polyhedron representation., 1972. 3

[6] Michael Bleyer, Christoph Rhemann, and Carsten Rother. Patchmatch stereo-stereo matching with slanted support windows. In *Bmvc*, 2011. 3

[7] Chris Buehler, Michael Bosse, Leonard McMillan, Steven Gortler, and Michael Cohen. Unstructured lumigraph rendering. In *SIGGRAPH*, 2001. 2

[8] Zhiqin Chen, Andrea Tagliasacchi, and Hao Zhang. Bsp-net: Generating compact meshes via binary space partitioning. *CVPR*, 2020. 3

[9] Paul Debevec, Yizhou Yu, and George Borshukov. Efficient view-dependent image-based rendering with projective texture-mapping. In *Eurographics Workshop on Rendering Techniques*. Springer, 1998. 1, 2

[10] Paul E Debevec, Camillo J Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs: A hybrid geometry-and image-based approach. In *SIGGRAPH*, 1996. 1, 2

[11] Maximilian Denninger, Martin Sundermeyer, Dominik Winkelbauer, Youssef Zidan, Dmitry Olefir, Mohamad Elbadrawy, Ahsan Lodhi, and Harinandan Katam. Blenderproc. 2019. 6

[12] John Flynn, Michael Broxton, Paul Debevec, Matthew DuVall, Graham Fyffe, Ryan Overbeck, Noah Snavely, and Richard Tucker. Deepview: View synthesis with learned gradient descent. In *CVPR*, 2019. 1, 2

[13] John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. Deepstereo: Learning to predict new views from the world's imagery. In *CVPR*, 2016. 1, 2

[14] David Ford Fouhey, Abhinav Gupta, and Martial Hebert. Unfolding an indoor origami world. In *ECCV*. Springer, 2014. 3

[15] Yasutaka Furukawa, Brian Curless, Steven M Seitz, and Richard Szeliski. Manhattan-world stereo. In *CVPR*, 2009. 3

[16] David Gallup, Jan-Michael Frahm, Philippos Mordohai, Qingxiong Yang, and Marc Pollefeys. Real-time plane-sweeping stereo with multiple sweeping directions. In *CVPR*. IEEE, 2007. 3

[17] Stephan J Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. Fastnerf: High-fidelity neural rendering at 200fps. *ICCV*, 2021. 3, 5

[18] Georgia Gkioxari, Jitendra Malik, and Justin Johnson. Mesh r-cnn. In *ICCV*, 2019. 3

[19] Steven J Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F Cohen. The lumigraph. In *SIGGRAPH*, 1996. 3

[20] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. A papier-mâché approach to learning 3d surface generation. In *CVPR*, 2018. 3

[21] Abhinav Gupta, Alexei A Efros, and Martial Hebert. Blocks world revisited: Image understanding using qualitative geometry and mechanics. In *ECCV*. Springer, 2010. 3

[22] Varsha Hedau, Derek Hoiem, and David Forsyth. Recovering the spatial layout of cluttered rooms. In *ICCV*, 2009. 3

[23] Varsha Hedau, Derek Hoiem, and David Forsyth. Thinking inside the box: Using appearance models and context based on room geometry. In *ECCV*. Springer, 2010. 3

[24] Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. Deep blending for free-viewpoint image-based rendering. *SIGGRAPH*, 2018. 2

[25] Peter Hedman, Pratul P Srinivasan, Ben Mildenhall, Jonathan T Barron, and Paul Debevec. Baking neural radiance fields for real-time view synthesis. *ICCV*, 2021. 3

[26] Ronghang Hu, Nikhila Ravi, Alexander C Berg, and Deepak Pathak. Worldsheet: Wrapping the world in a 3d sheet for view synthesis from a single image. In *ICCV*, 2021. 3

[27] Michal Irani and Prabu Anandan. Parallax geometry of pairs of points for 3d scene analysis. In *ECCV*. Springer, 1996. 3

[28] Takeo Kanade. A theory of origami world. *Artificial intelligence*, 1980. 3

[29] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2015. 6

[30] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *SIGGRAPH*, 2017. 4, 6

[31] Johannes Kopf, Michael F Cohen, and Richard Szeliski. First-person hyper-lapse videos. *SIGGRAPH*, 2014. 1, 2

[32] David C Lee, Martial Hebert, and Takeo Kanade. Geometric reasoning for single image structure recovery. In *CVPR*. IEEE, 2009. 3

[33] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. *ICCV*, 2021. 3

[34] David B Lindell, Julien NP Martel, and Gordon Wetzstein. Autoint: Automatic integration for fast neural volume rendering. In *CVPR*, 2021. 3

[35] Chen Liu, Kihwan Kim, Jinwei Gu, Yasutaka Furukawa, and Jan Kautz. Planercnn: 3d plane detection and reconstruction from a single image. In *CVPR*, 2019. 2, 3

[36] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *NeurIPS*, 2020. 3, 6

[37] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *SIGGRAPH*, 2019. 3, 6

[38] Stephen Lombardi, Tomas Simon, Gabriel Schwartz, Michael Zollhoefer, Yaser Sheikh, and Jason Saragih. Mixture of volumetric primitives for efficient neural rendering. *SIGGRAPH*, 2021. 3

[39] Arun Mallya, Ting-Chun Wang, Karan Sapra, and Ming-Yu Liu. World-consistent video-to-video synthesis. In *ECCV*. Springer, 2020. 1, 2

[40] David Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. 1982. 3

[41] Ricardo Martin-Brualla, Rohit Pandey, Sofien Bouaziz, Matthew Brown, and Dan B Goldman. Gelato: Generative latent textured objects. In *ECCV*. Springer, 2020. 2

[42] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *CVPR*, 2021. 3

[43] 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. *SIGGRAPH*, 2019. 2

[44] 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, 6, 7

[45] Thomas Neff, Pascal Stadlbauer, Mathias Parger, Andreas Kurz, Chakravarty R Alla Chaitanya, Anton Kaplanyan, and Markus Steinberger. Donerf: Towards real-time rendering of neural radiance fields using depth oracle networks. *Computer Graphics Forum (EGSR)*, 2021. 3

[46] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *CVPR*, 2019. 2

[47] Martin Piala and Ronald Clark. Terminerf: Ray termination prediction for efficient neural rendering. *3DV*, 2021. 3

[48] Yiming Qian and Yasutaka Furukawa. Learning pairwise inter-plane relations for piecewise planar reconstruction. In *ECCV*. Springer, 2020. 2

[49] Daniel Rebain, Wei Jiang, Soroosh Yazdani, Ke Li, Kwang Moo Yi, and Andrea Tagliasacchi. Derf: Decomposed radiance fields. In *CVPR*, 2021. 3

[50] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. *ICCV*, 2021. 1, 3, 5, 6, 7

[51] Gernot Riegler and Vladlen Koltun. Free view synthesis. In *ECCV*. Springer, 2020. 1, 2

[52] Gernot Riegler and Vladlen Koltun. Stable view synthesis. In *CVPR*, 2021. 1, 2

[53] Darius Rückert, Linus Franke, and Marc Stamminger. Adop: Approximate differentiable one-pixel point rendering. *in ArXiv*, 2021. 1, 2

[54] Renato F Salas-Moreno, Ben Glocken, Paul HJ Kelly, and Andrew J Davison. Dense planar slam. In *ISMAR*. IEEE, 2014. 3

[55] Harpreet S Sawhney. 3d geometry from planar parallax. In *CVPR*, 1994. 3

[56] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 5, 6, 7

[57] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, 2016. 6, 7

[58] Jonathan Shade, Steven Gortler, Li-wei He, and Richard Szeliski. Layered depth images. In *SIGGRAPH*, 1998. 2

[59] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. *NeurIPS*, 2019. 6

[60] Pratul P Srinivasan, Richard Tucker, Jonathan T Barron, Ravi Ramamoorthi, Ren Ng, and Noah Snavely. Pushing the boundaries of view extrapolation with multiplane images. In *CVPR*, 2019. 2

[61] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. *in ArXiv*, 2019. 4, 6

[62] Shih-Yang Su, Frank Yu, Michael Zollhoefer, and Helge Rhodin. A-nerf: Surface-free human 3d pose refinement via neural rendering. *NeurIPS*, 2021. 5

[63] Richard Szeliski, P Anandan, and Simon Baker. From 2d images to 2.5 d sprites: A layered approach to modeling 3d scenes. In *Proceedings IEEE International Conference on Multimedia Computing and Systems*. IEEE, 1999. 1, 3

[64] Alejandro Sztrajman, Gilles Rainer, Tobias Ritschel, and Tim Weyrich. Neural brdf representation and importance sampling. In *Computer Graphics Forum*. Wiley Online Library, 2021. 3

[65] Matthew Tancik, Pratul P Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *NeurIPS*, 2020. 4

[66] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *SIGGRAPH*, 2019. 2

[67] Justus Thies, Michael Zollhöfer, Christian Theobalt, Marc Stamminger, and Matthias Nießner. Image-guided neural object rendering. In *ICLR*. OpenReview. net, 2020. 2

[68] Richard Tucker and Noah Snavely. Single-view view synthesis with multiplane images. In *CVPR*, 2020. 2

[69] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *CVPR*, 2021. 3

[70] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *TIP*, 2004. 6

[71] Yi Wei, Shaohui Liu, Yongming Rao, Wang Zhao, Jiwen Lu, and Jie Zhou. Nerfingmvs: Guided optimization of neural radiance fields for indoor multi-view stereo. In *ICCV*, 2021. 3, 7

[72] Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. Synsin: End-to-end view synthesis from a single image. In *CVPR*, 2020. 2

[73] Suttisak Wizadwongsa, Pakkapon Phongthawee, Jiraphon Yenphraphai, and Supasorn Suwajanakorn. Nex: Real-time view synthesis with neural basis expansion. In *CVPR*, 2021. 1, 2, 4, 5, 6, 7

[74] Fanbo Xiang, Zexiang Xu, Milos Hasan, Yannick Hold-Geoffroy, Kalyan Sunkavalli, and Hao Su. Neutex: Neural texture mapping for volumetric neural rendering. In *CVPR*, 2021. 2

[75] Zhenpei Yang, Yuning Chai, Dragomir Anguelov, Yin Zhou, Pei Sun, Dumitru Erhan, Sean Rafferty, and Henrik Kretzschmar. Surfelgan: Synthesizing realistic sensor data for autonomous driving. In *CVPR*, 2020. 2

[76] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *in ArXiv*, 2021. 2

[77] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Ronen Basri, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. *NeurIPS*, 2020. 2

[78] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. *ICCV*, 2021. 1, 3, 5, 6, 7

[79] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *CVPR*, 2021. 1, 3

[80] Zehao Yu, Jia Zheng, Dongze Lian, Zihan Zhou, and Shenghua Gao. Single-image piece-wise planar 3d reconstruction via associative embedding. In *CVPR*, 2019. 3

[81] Jason Y Zhang, Gengshan Yang, Shubham Tulsiani, and Deva Ramanan. Ners: Neural reflectance surfaces for sparse-view 3d reconstruction in the wild. *NeurIPS*, 2021. 2

[82] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *in ArXiv*, 2020. 1, 3, 7, 8

[83] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 6

[84] Xiuming Zhang, Pratul P Srinivasan, Boyang Deng, Paul Debevec, William T Freeman, and Jonathan T Barron. Nerfactor: Neural factorization of shape and reflectance under an unknown illumination. *SIGGRAPH*, 2021. 1, 3

[85] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: learning view synthesis using multiplane images. *SIGGRAPH*, 2018. 1, 2