Multi-Space Neural Radiance Fields

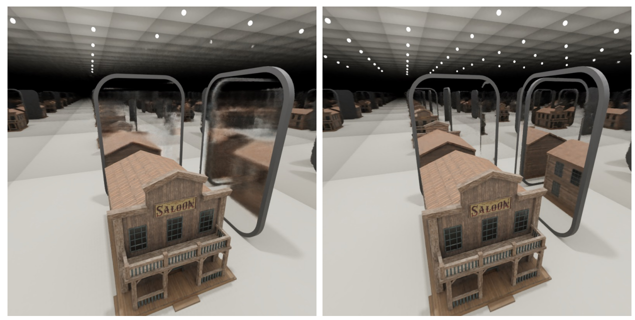
Ze-Xin Yin Jiaxiong Qiu Ming-Ming Cheng Bo Ren*
VCIP, CS, Nankai University
{Zexin.Yin.cn, qiujiaxiong727}@gmail.com, {cmm, rb}@nankai.edu.cn

Abstract

Existing Neural Radiance Fields (NeRF) methods suffer from the existence of reflective objects, often resulting in blurry or distorted rendering. Instead of calculating a single radiance field, we propose a multi-space neural radiance field (MS-NeRF) that represents the scene using a group of feature fields in parallel sub-spaces, which leads to a better understanding of the neural network toward the existence of reflective and refractive objects. Our multi-space scheme works as an enhancement to existing NeRF methods, with only small computational overheads needed for training and inferring the extra-space outputs. We demonstrate the superiority and compatibility of our approach using three representative NeRF-based models, i.e., *NeRF, Mip-NeRF, and Mip-NeRF 360. Comparisons are performed on a novelly constructed dataset consisting of 25 synthetic scenes and 7 real captured scenes with complex reflection and refraction, all having 360-degree viewpoints. Extensive experiments show that our approach significantly outperforms the existing single-space NeRF methods for rendering high-quality scenes concerned with complex light paths through mirror-like objects. Our code and dataset will be publicly available at https://zx-yin.github.io/msnerf.*

1. Introduction

Neural Radiance Fields (NeRF) [25] and its variants are refreshing the community of neural rendering, and the potential for more promising applications is still under exploration. NeRF represents scenes as continuous radiance fields stored by simple Multi-layer Perceptrons (MLPs) and renders novel views by integrating the densities and radiance, which are queried from the MLPs by points sampled along the ray from the camera to the image plane. Since its first presentation [25], many efforts have been investigated to enhance the method, such as extending to unbounded scenes [2, 50], handling moving objects [29, 30, 37], or reconstructing from pictures in the wild [6, 21, 35, 49].



(a) Mip-NeRF 360 [2], SSIM=0.825 (b) Our Model, SSIM=0.881

Figure 1. (a) Though Mip-NeRF 360 can handle unbounded scenes, it still suffers from reflective surfaces, as the virtual images violate the multi-view consistency, which is of vital importance to NeRF-based methods. (b) Our method can help conventional NeRF-like methods learn the virtual images with little extra cost.

However, rendering scenes with mirrors is still a challenging task for state-of-the-art NeRF-like methods. One of the principle assumptions for the NeRF method is the multi-view consistency property of the target scenes [16, 20, 36]. When there are mirrors in the space, if one allows the viewpoints to move 360-degree around the scene, there is no consistency between the front and back views of a mirror, since the mirror surface and its reflected virtual image are only visible from a small range of views. As a result, it is often required to manually label the reflective surfaces in order to avoid falling into sub-optimal convergences [12].

In this paper, we propose a novel multi-space NeRF-based method to allow the automatic handling of mirror-like objects in the 360-degree high-fidelity rendering of scenes without any manual labeling. Instead of regarding the Euclidean scene space as one single space, we treat it as composed of multiple virtual sub-spaces, whose composition changes according to location and view direction. We show that our approach using such a multi-space decomposition leads to successful handlements of complex reflections and refractions where the multi-view consistency is heavily violated in the Euclidean real space. Furthermore, we show that the above benefits can be achieved by designing a low-cost multi-space module and replacing the original output layer with it. Therefore, our multi-space approach serves as

*Bo Ren is the corresponding author.

a general enhancement to the NeRF-based backbone, equipping most NeRF-like methods with the ability to model complex reflection and refraction, as shown in Fig. 1.

Existing datasets have not paid enough attention to the 360-degree rendering of scenes containing mirror-like objects, such as RFFR [12] just has forward-facing scenes, and the Shiny dataset in [42] with small viewpoints changes and cannot exhibit view-dependent effects in large angle scale. Therefore we construct a novel dataset dedicated to evaluation for the 360-degree high-fidelity rendering of scenes containing complex reflections and refractions. In this dataset, we collect 25 synthesized scenes and 7 captured real-world scenes. Each synthesized scene consists of 120 images of 360-degree around reflective or refractive objects, with 100 randomly split for training, 10 for validation, and 10 for evaluation. Each real-world scene is captured randomly around scenes with reflective and refractive objects, consisting of 62 to 118 images, and organized under the convention of LLFF [24]. We then demonstrate the superiority and compatibility of our approach by comparisons, using three representative baseline models, *i.e.*, NeRF [25], Mip-NeRF [1], and Mip-NeRF 360 [2], with and without our multi-space module. Experiments show that our approach improves performance by a large margin both quantitatively and qualitatively on scenes with reflection and refraction. Our main contributions are as follows:

- We propose a multi-space NeRF method that automatically handles mirror-like objects in 360-degree high-fidelity scene rendering, achieving significant improvements over existing representative baselines both quantitatively and qualitatively.
- We design a lightweight module that can equip most NeRF-like methods with the ability to model reflection and refraction with small computational overheads.
- We construct a dataset dedicated to evaluation for the 360-degree high-fidelity rendering of scenes containing complex reflections and refractions, including 25 synthesized scenes and 7 real captured scenes.

2. Related works

Coordinate-based novel view synthesis. NeRF [25] has bridged the gap between computer vision and computer graphics, and reveals a promising way to render high-quality photorealistic scenes with only posed images. The insights and the generalization ability of this scheme also facilitate various tasks both in CV and CG, *i.e.*, 3D reconstruction [28,40], 3D-aware generation [4,15,27], 3D-aware edition [39, 47], and avatar reconstruction and manipulation [9, 18, 52]. Besides, researchers have made great efforts to enhance this scheme. Mip-NeRF [1] enhances the anti-aliasing ability of NeRF by featuring 3D conical frustum using integrated positional encoding. [14,23] adapt this scheme to HDR images. [2,50] extend NeRF and its variants to unbounded scenes. There are also many works trying to speed up the training and inference speed using explicit or hybrid representations [5, 7, 10, 26, 32, 34, 46].

Glossy materials with high specular have a great influence on NeRF-like methods, [38] is inspired by precomputation-based techniques [31] in computer graphics to represent and render view-dependent specular and reflection, but it fails to handle mirror-like reflective surfaces because the virtual images cannot be treated as textures. Guo *et al*. [12] propose to decompose reflective surfaces into a transmitted part and reflected part, which is the most relevant work to ours, but such decomposition cannot handle 360-degree views with mirror-like objects because the virtual images have no difference from real objects until the viewpoint moves beyond a certain angle.

Another line of work similar to ours is multiple neural radiance fields, but they do so for different purposes [11,27, 32, 43, 44]. [27] uses object-level neural radiance fields for 3D-aware generation and composition. [32, 44] uses multiple small MLPs for efficient rendering. [11, 43] uses multiple object-level neural radiance fields for 3D scene decomposition and edition.

Commonly used datasets. Researchers have introduced or constructed many different datasets to facilitate the development of NeRF-based methods in various tasks. Mildenhall *et al*. [25] collect a dataset containing eight rendered sets of posed images about eight objects separately, and eight real captured forward-facing scenes with the camera poses and intrinsics estimated by COLMAP [33]. Nevertheless, these scenes lack reflection and refraction, which are very common. Wizadwongsa *et al*. [42] propose a dataset, namely Shiny, that contains eight more challenging scenes to test NeRF-like methods on view-dependent effects, but they are captured in a roughly forward-facing manner. Verbin *et al*. [38] create a dataset of six glossy objects, namely Shiny Blender, which are rendered under similar conditions as done in NeRF to test methods in modeling more complex materials. For unbounded scenes, Barron *et al*. [2] construct a dataset consisting of 5 outdoor scenes and 4 indoor scenes, while Zhang *et al*. [50] adopt Tanks and Temples (T&T) dataset [19] and the Light Field dataset [48]. Bemana1 *et al*. [3] capture a dataset consisting of refractive objects, which is composed of four scenes with cameras moving in a large range. Guo *et al*. [12] collect six forward-facing scenes with reflective and semi-transparent materials, which is, to date, the most relevant dataset to ours, but ours is much more challenging. DTU dataset [17] and Blended-MVS dataset [45] are commonly used as benchmarks for the evaluation of 3D reconstruction.

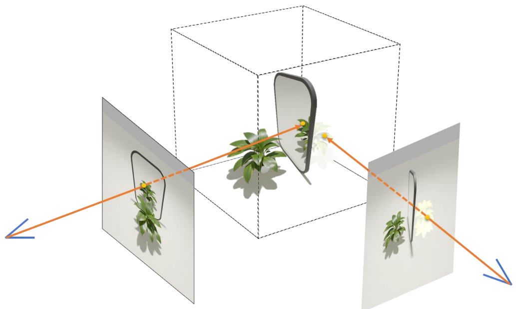


Figure 2. The virtual image created by the mirror is visible only in a small range of views, which violates the multi-view consistency.

3. Method

3.1. Preliminaries: Neural Radiance Fields

Neural Radiance Fields (NeRF) [25] encodes a scene in the form of continuous volumetric fields into the weights of a multilayer perceptron (MLP), and adopts the absorption-only model in the traditional volumetric rendering to synthesize novel views. The training process only requires a sparse set of posed images and casts rays $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ through the scene, where $\mathbf{o} \in \mathbb{R}^3$ is the camera center and $\mathbf{d} \in \mathbb{R}^3$ is the view direction, and the rays can be calculated by intrinsics and poses from the training data. Given these rays, NeRF samples a set of 3D points $\{\mathbf{p}_i = \mathbf{o} + t_i\mathbf{d}\}$ by the distance to the camera t_i in the Euclidean space and projects these points to a higher dimensional space using the following function:

$$\gamma(\mathbf{p}) = [\sin(\mathbf{p}), \cos(\mathbf{p}), ..., \sin(2^{L-1}\mathbf{p}), \cos(2^{L-1}\mathbf{p})] \quad (1)$$

where L is a hyperparameter and $\mathbf{p}$ is a sampled point.

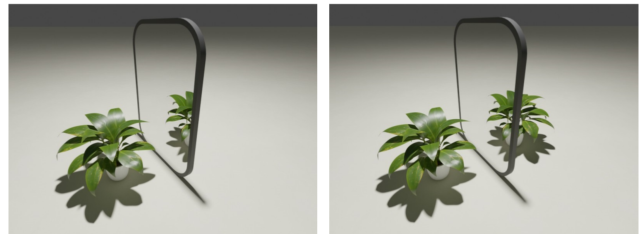
Given the projected features $\{\gamma(\mathbf{p}_\mathrm{i})\}$ and the ray direction $\mathbf{d}$, the MLP outputs the densities $\{\sigma_i\}$ and colors $\{\mathbf{c}_i\}$, which are used to estimate the color $\mathbf{C}(\mathbf{r})$ of the ray using the quadrature rule reviewed by Max [22]:

$$\hat{\mathbf{C}}(\mathbf{r}) = \sum_{i=1}^{N} T_i(1 - \exp(-\sigma_i\delta_i))\mathbf{c}_i \quad (2)$$

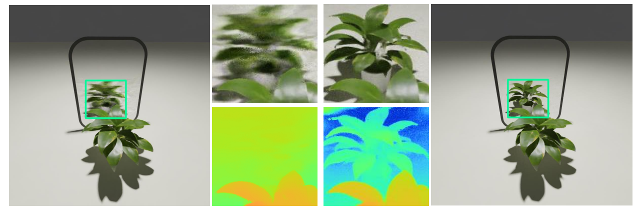
with $T_i = \exp(-\sum_{j=1}^{i-1} \sigma_j\delta_j)$ and $\delta_i = t_i - t_{i-1}$. Since the equation is differentiable, the model parameters can be optimized directly by Mean Squared Error (MSE) loss:

$$\mathcal{L} = \frac{1}{|\mathcal{R}|} \sum_{r\in\mathcal{R}} ||\hat{\mathbf{C}}(\mathbf{r}) - \mathbf{C}(\mathbf{r})||_2 \quad (3)$$

where $\mathcal{R}$ is a training batch of rays. Besides, NeRF also adopts a hierarchical sampling strategy to sample more points where higher weights are accumulated. With these designs, NeRF achieves state-of-the-art photorealistic results of novel view synthesis in most cases.



(a) A training view in toy scene A. (b) A training view in toy scene B.



(c) A render view in toy scene A. (d) A render view in toy scene B.

Figure 3. The first row is training view examples in the two scenes. In scene A there is only a plant in front of a mirror, while in scene B we carefully place another plant to match the exact position where the virtual image lies. The second row is test views with rendered depth from the vanilla NeRF trained on the toy scenes. As demonstrated, NeRF can avoid the trap of treating reflected images as textures when the 'virtual image' satisfies multi-view consistency.

3.2. Multi-space Neural Radiance Field

The volumetric rendering equation and the continuous representation ability of MLPs do guarantee the success of NeRF-based methods in novel view synthesis, but as pointed out by previous works [12, 16, 20], there is also an unignorable property hidden in the training process that helps the convergence, which is the multi-view consistency. However, the multi-view consistency can be easily violated by any reflective surfaces. An example is shown in Fig. 2, when looking in front of a mirror one can observe the reflective virtual image as if there were an object behind it, but when looking from a side or backward, there is actually nothing behind the mirror. In practice, this means there will be completely conflictive training batches violating the fitting process of MLP.

To experimentally demonstrate the importance of multi-view consistency and its influence on the conventional NeRF network structure, We create two 360-degree toy scenes using an open source software Blender [8], each of which consists of 100 training images and 10 test images, training view examples are shown in Fig. 3a and Fig. 3b. The only difference between the two scenes is that we place a mirror-posed real object behind the mirror in the latter scene, but not in the former one. We train the vanilla NeRF separately on these toy scenes under the same setting and render some views from the test set as in Fig. 3c and Fig. 3d, which clearly shows that the vanished virtual image (*i.e.*, violation to the multi-view consistency) in some views leads

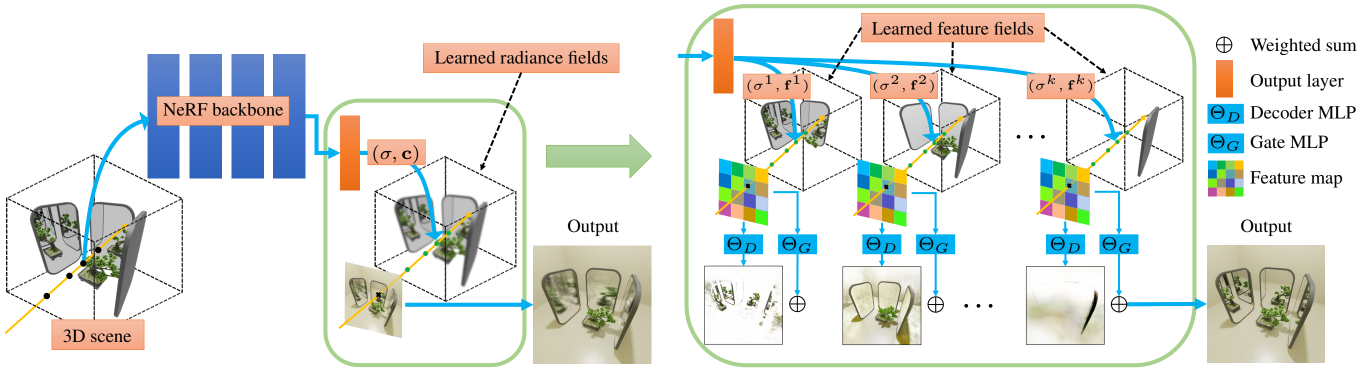


Figure 4. Our multi-space module only modifies the output and volumetric rendering part of the network. The original NeRF calculates a pair of density σ and radiance $\mathbf{c}$ to get the accumulated color. Our output layer produces pairs of densities $\{\sigma^k\}$ and features $\{\mathbf{f}^k\}$, which correspond to multiple parallel feature fields. Then, we use volumetric rendering to get multiple feature maps. Two simple MLPs, *i.e.*, Decoder MLP and Gate MLP, are utilized to decode RGB maps and pixel-wise weights from these feature maps.

the model to suboptimal results in reflection-related regions and produces blur in rendering. Interestingly, the conventional NeRF is still trying to fulfill the multi-view consistency assumption in the process. From the depth map in Fig. 3c, we can easily conclude that the conventional NeRF treats the viewed virtual image as a "texture" on the reflective surface, achieving a compromise between its principle assumption and the conflicts in training data, although the compromise leads to false understandings and worse rendering results of the real scenes.

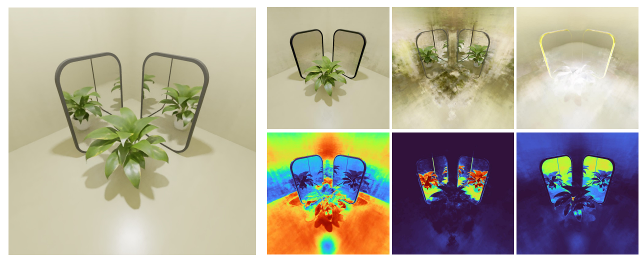
Contrary to the conventional NeRF, inspired by the common perspective in Physics and Computer Graphics that reflective light can be viewed as "directly emitted" from its mirror-symmetric direction, from a possible "virtual source inside the virtual space in the mirror," we build our novel multi-space NeRF approach on the following assumption:

Assumption 1 *At the existence of reflection and refraction, the real Euclidean scene space can be decomposed into multiple virtual sub-spaces. Each sub-space satisfies the multi-view consistency property.*

It follows that the composition weights of the sub-spaces can change according to the spatial location and the view direction. Thus all sub-space contributes dynamically to the final render result. In this way, the violation of the multi-view consistency in real Euclidean space when there is a reflective surface can be overcome by placing the virtual images in certain sub-spaces only visible from certain views, as shown in Fig. 5.

3.3. Multi-Space module

A naive implementation of the multi-space NeRF network would be constructing the network using multiple tiny parallel MLPs, with each one representing one sub-space information, which, however, will largely increase parameters and has been proven to be tough to converge [32].



(a) Composed render result. (b) RGB and weights of sub-spaces.

Figure 5. We visualize a novel view and a few decoded images with the corresponding weights of some sub-spaces from our MS-NeRF$_B$ model in Sec. 5.2. The results show that our method successfully decomposes virtual images into certain sub-spaces.

Besides, our experiments in Sec. 3.2 demonstrate that the current network structure of NeRF possesses the potential to understand our 3D scenes. Therefore, we propose a compact Multi-Space module (MS module), which takes advantage of the neural feature field scheme originally designed for memory saving [27], to sufficiently extract multi-space information from standard NeRF backbone network structures with only small computational overheads. Specifically, the MS module will replace the original output layer of the NeRF backbone. Below we describe the detailed architecture of our module.

As shown in Fig. 4, our MS module only modifies the output part of vanilla NeRF. Vanilla NeRF computes single density σ_i and radiance $\mathbf{c}_i$ for each position along a ray casting through the scene and performs volumetric rendering using Eq. (2) to get the accumulated color. On the contrary, our multi-head layer replaces the neural radiance field with the neural feature fields [27]. Specifically, the modified output layer gives K densities $\{\sigma_i^k\}$ and features $\{\mathbf{f}_i^k\}$ of d dimension for each position along a ray with each pair corresponding to a sub-space, where K and d are hyper-

dataset	origin	applications	Type	viewpoints	properties	number
Realistic Synthetic 360°	[25]	view synthesis	S	360-degree	non-Lambertian	8
Real Forward-Facing	[24, 25]	view synthesis	R	forward-facing	non-Lambertian	8
Shiny	[42]	view synthesis	R	forward-facing	high-specular, refraction	8
Tanks and Temples(T&T)	[19]	view synthesis	R	360-degree	unbounded scenes	4
Mip-NeRF 360	[2]	view synthesis	R	360-degree	unbounded scenes	9
EikonalFields	[3]	view synthesis	R	large angle view	refraction	4
RFFR	[12]	view synthesis	R	forward-facing	reflection, semi-transparent	6
DTU	[17]	reconstruction	R	360-degree	non-Lambertian objects	15*
BlendedMVS	[45]	reconstruction	S	360-degree	non-Lambertian scenes	7*
Shiny Blender	[38]	view synthesis	S	360-degree	glossy materials	6
Ref-NeRF Real captured scenes	[13, 38]	view synthesis	R	360-degree	glossy materials	3

Table 1. Properties of a commonly used dataset for NeRF-based methods. 'S' and 'R' represent synthesized and real captured, respectively. We denote those unnamed datasets with the name of the methods. '*' refers to the number of scenes commonly used by NeRF-based methods, as the original dataset contains more scenes than noted, and we do not take them into consideration.

parameters for the total sub-space number and the feature dimension of the neural feature field, respectively.

We then integrate features along the ray in each sub-space to collect K feature maps that encode the color information and visibility of each sub-space from a certain viewpoint. As all pixels are calculated the same way, we denote each pixel as $\{\mathcal{F}^k\}$ for simplicity and describe the operation at the pixel level. Each pixel $\{\mathcal{F}^k\}$ of the feature maps is calculated using:

$$\hat{\mathcal{F}}^k(\mathbf{r}) = \sum_{i=1}^{N} T_i^k(1 - \exp(-\sigma_i^k \delta_i))\mathbf{f}_i^k, \tag{4}$$

where the superscript k indicates the sub-space that the ray casts through. The k-th density σ_i^k and feature $\mathbf{f}_i^k$ correspond to the k-th sub-space. $T_i^k = \exp(-\sum_{j=1}^{i-1} \sigma_j^k \delta_j)$ and $\delta_i = t_i - t_{i-1}$ are similarly computed as in Eq. (2).

Then, $\{\mathcal{F}^k\}$ is decoded by two small MLPs, each with just one hidden layer. The first is a Decoder MLP that takes $\{\mathcal{F}^k\}$ as input and outputs RGB vectors. The second is a Gate MLP that takes $\{\mathcal{F}^k\}$ as input and outputs weights that control the visibility of certain sub-space. Specifically, we use:

$$\{\mathcal{F}^k\} \xrightarrow{\Theta_D} \{\mathbf{C}^k\}, \{\mathcal{F}^k\} \xrightarrow{\Theta_G} \{w^k\}, \tag{5}$$

where Θ_D represents the Decoder MLP, and Θ_G represents the Gate MLP. In the end, the MS module applies the softmax function to $\{w^k\}$ as the color contribution of each sub-space to form the final render results:

$$\hat{\mathbf{C}}(\mathbf{r}) = \frac{1}{\sum_{i=1}^{K} \exp(w^i)} \sum_{k=1}^{K} \exp(w^k)\mathbf{C}^k. \tag{6}$$

Eq. (6) needs no additional loss terms compared with the vanilla NeRF method. As a result, the above light-weighted MS module is able to serve as an enhancement module onto the conventional NeRF-like backbone networks, and we will show that our approach achieves significant enhancements in Sec. 5.2.

4. Dataset

4.1. Existing datasets

We briefly revisit the commonly used or most relevant datasets to our task and list their properties in Tab. 1. As can be seen, there lacks a 360-degree benchmark for scenes with complex light paths, *e.g.*, a glass of water in front of a mirror.

4.2. Our proposed dataset

As summarized in Sec. 4.1, there lacks a 360-degree dataset consisting of complex reflection and refraction to facilitate the related research. Therefore we collect a 360-degree dataset comprising 25 synthetic scenes and 7 real captured scenes.

For our synthesized part shown in Fig. 6a, we use an open source software Blender [8] and design our scenes with 3D models from BlenderKit, a community for sharing 3D models. As our dataset consists of complete scenes instead of single objects, we fix the height of our camera position with the camera looking at the center of the scene and moving the camera around a circle to render the whole scene.

For all our scenes, we uniformly sample 120 points along the circle and randomly choose 100 images for the training set, 10 for the validation set, and 10 for the test set. The constructed dataset features a wide variety of scenes containing reflective and refractive objects. We include a variety of complexity of light paths, controlled by the number and the layout of the mirror(s) in the scene, where the number of mirrors ranges from 1 up to tens of small pieces. Note that even a scene in our dataset with only one mirror is more challenging than RFFR [25], as our camera moves from the

(a) A part of our synthesized dataset.

(b) A part of our real captured dataset.

Figure 6. Demo scenes of our datasets (more in the supplementary). Our dataset exhibits diversities of reflection and refraction, which can serve as a benchmark for validating the ability to synthesize novel views with complex light paths.

front to the back of the mirror(s). Besides, we also construct rooms with mirror walls that can essentially be treated as unbounded scenes, where we add mirrors in the center of the room and create unbounded virtual images. We further build challenging scenes, including a combination of reflection and refraction.

We also include 7 captured real scenes with complex light conditions shown in Fig. 6b. We construct our scenes using two mirrors, one glass ball with a smooth surface, one glass ball with a diamond-like surface, a few toys, and a few books. We capture pictures randomly with 360-degree viewpoints.

5. Experiments

5.1. Hyperparameters and benchmarks

We conduct three sets of experiments based on different datasets with different baselines and our modules of different scales. As our module is quite simple, we can scale our module by three hyperparameters, which we refer to as K for the sub-space number, d for the dimension of output features, and h for the hidden layer dimension of Decoder MLP and Gate MLP, respectively. To compare fairly, we carry out all the experiments following most default setting from [1,2,12,25,38], except that we use 1024 rays per batch and train 200k iterations for all experiments on all scenes. Experiment details are as follows.

We select three representative NeRF-based models as our baselines and integrate our modules with them. For NeRF [25] and Mip-NeRF [1] based experiments, we build MS-NeRF$_S$ and MS-Mip-NeRF$_S$ with hyperparameters $\{K = 6, d = 24, h = 24\}$, similarly MS-NeRF$_M$ and MS-Mip-NeRF$_M$ with hyperparameters $\{K = 6, d = 48, h = 48\}$, and MS-NeRF$_B$ and MS-Mip-NeRF$_B$ with hyperparameters $\{K = 8, d = 64, h = 64\}$. For Mip-NeRF 360 [2] based experiments, we construct MS-Mip-NeRF 360 with hyperparameters $\{K = 8, d = 32, h = 64\}$. Moreover, we provide a comparison with Ref-NeRF [38] because it uses Mip-NeRF as a baseline and possesses an outstanding ability to model glossy materials.

	PSNR↑	SSIM↑	LPIPS↓	# Params
NeRF	30.82	0.865	0.209	1.159M
MS-NeRF$_S$	32.39	0.872	0.201	1.201M
MS-NeRF$_M$	32.61	0.875	0.195	1.245M
MS-NeRF$_B$	32.77	0.876	0.195	1.311M
Mip-NeRF	31.42	0.874	0.215	0.613M
Ref-NeRF	32.37	0.882	0.180	0.713M
MS-Mip-NeRF$_S$	33.63	0.886	0.195	0.634M
MS-Mip-NeRF$_M$	33.80	0.887	0.193	0.656M
MS-Mip-NeRF$_B$	33.90	0.888	0.191	0.689M
Mip-NeRF 360	31.58	0.895	0.145	9.007M
MS-Mip-NeRF 360	35.04	0.906	0.130	9.052M

(a) Comparisons on the synthetic part of our dataset.

	PSNR↑	SSIM↑	LPIPS↓	# Params
Mip-NeRF 360	26.70	0.889	0.113	9.007M
MS-Mip-NeRF 360	28.14	0.891	0.119	9.052M

(b) Comparisons on the real captured part of our dataset.

	PSNR↑	SSIM↑	LPIPS↓	# Params
NeRFReN*	35.26	0.940	0.081	1.264M
MS-NeRF$_T$	35.93	0.948	0.066	1.295M

(c) Comparisons on RFFR dataset. '*' denotes that we re-train the model using the official code following the provided setting, except the number of masks used for reflective surfaces is 0.

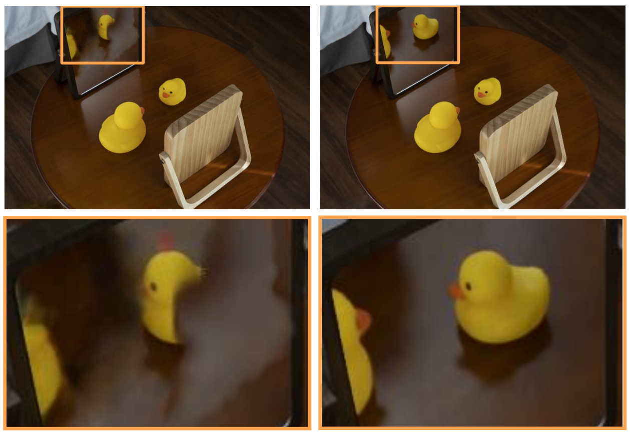
Table 2. Quantitative comparisons with existing methods.

We also compare our method with NeRFReN on the RFFR dataset [12]. NeRFReN is a specially designed two-branch network based on vanilla NeRF for mirror-like surfaces in forward-facing scenes. Thus we construct a tiny version of our method, referred to as MS-NeRF$_T$, based on NeRF with hyperparameters $\{K = 2, d = 128, h = 128\}$. Here we use two sub-spaces as NeRFReN tries to decompose reflective surfaces into two parts, and we want to show that our space decomposition is more effective. For a fair comparison, we re-train NeRFReN using the official code under the provided settings on the RFFR dataset, except that we set the number of the used mask to 0 as our method requires no extra mask.

All the training details can be found in the supplementary. We report our results with three commonly used metrics: PSNR, SSIM [41], and LPIPS [51].

(a) Mip-NeRF 360 (b) MS-Mip-NeRF 360

Figure 7. Visual comparison between Mip-NeRF 360 and MS-Mip-NeRF 360. Our module can extend Mip-NeRF 360 to model unbounded virtual scenes.



(a) Mip-NeRF 360 (b) MS-Mip-NeRF 360

Figure 8. Visual comparison between Mip-NeRF 360 and MS-Mip-NeRF 360 on the real captured part of our dataset. Our method is robust enough to recover virtual images in the real world.

5.2. Comparisons

Quantitative comparisons. As reported in Tab. 2a, our module can be integrated into most NeRF-like models and improve performance by a large margin with minimal extra cost introduced. Especially in Mip-NeRF 360-based experiments, our module exhibits better results of 3.46 dB improvement in PSNR with merely 0.5% extra parameters. Besides, our Mip-NeRF-based models also outperform Ref-NeRF [38] by a large margin, which is a variant based on Mip-NeRF with the outstanding ability to model glossy materials. We also demonstrate our results compared with the state-of-the-art Mip-NeRF 360 results on the real-captured part of our dataset in Tab. 2b. Our approach also shows large improvements. As shown in Tab. 2c, our approach achieves better results when no manually-labeled masks are provided in training on the RFFR dataset, which contains forward-facing reflective surfaces in the scenes. The above experiments demonstrate the superiority and compatibility of our method.

Qualitative comparisons and discussions. Besides quantitative comparisons, we summarize the advantages of our modules and support them by qualitatively or quantitatively comparing our methods with the corresponding baselines.

(a) Ref-NeRF (b) MS-Mip-NeRF$_B$

Figure 9. Visual comparison between MS-Mip-NeRF$_B$ and Ref-NeRF. Our method significantly outperforms Ref-NeRF on reflective surfaces.

(a) NeRFReN (b) MS-NeRF$_B$

Figure 10. (a) Trained with accurately labeled masks, NeRFReN even fails to render ordinary parts of the scene in 360-degree scenes with mirrors. (b) Our method requires no extra manually labeled masks and renders high-quality images.

Qualitative comparisons with the state-of-the-art method (*i.e.*, Mip-NeRF 360) are shown in Fig. 1, Fig. 7, and Fig. 8. Our method renders high-fidelity virtual images, bounded and unbounded, in both synthetic and real-world scenes.

A qualitative comparison with Ref-NeRF [38], which understands virtual images as textures using the conventional NeRF backbone, is shown in Fig. 9. As Ref-NeRF is also based on Mip-NeRF, we compare our Mip-NeRF-based variant with Ref-NeRF using the same baseline and use comparable parameters (specifically ours 0.689M and Ref-NeRF 0.713M) in the comparison. Again the qualitative results show our significant improvements in rendering reflective surfaces.

We also compare with the NeRFReN model, which requires accurately labeled masks of the reflective regions during training and handles forward-facing reflective surfaces only. In this comparison, we train their model on our synthesized dataset with extra accurate reflection masks provided. Fig. 10 shows that their model fails to recover 360-degree high-fidelity rendering while our approach succeeds.

5.3. Ablation studies

In this section, we evaluate the design of our module and explore the relation between the number of sub-space and the number of virtual images.

Overview Ground Truth MS-NeRF$_S$ MS-NeRF$_{Avg}$ NeRF

Figure 11. Comparing NeRF, MS-NeRF$_{Avg}$ and MS-NeRF$_S$.

	PSNR↑	SSIM↑	LPIPS↓	# Params
NeRF	30.82	0.866	0.209	1.159M
MS-NeRF$_{Avg}$	31.45	0.868	0.205	1.166M
MS-NeRF$_S$	32.39	0.872	0.201	1.201M

Table 3. Ablation study on our module architecture.

Ablation on using neural feature field. We implement a module that simply outputs K scalars $\{\sigma_i^k\}$ and K RGB vectors $\{\mathbf{c}_i^k\}$ of three dimensions, then we use the same integral equation as NeRF to get the pixel color of each sub-space and we average among all sub-spaces to get the final pixel color. We integrate this design into vanilla NeRF noted as MS-NeRF$_{Avg}$, where we set $K = 6$, and the results are reported in Tab. 3. We also exhibit a few visual results in Fig. 11, which indicate that a simple multi-space radiance field assumption can help the model partially overcome the violation of reflections, but will also introduce the over-smoothing problem because of the lack of an efficient multi-space composition strategy.

Ablation on the sub-space number. In our Euclidean space, one can control the number of virtual sub-spaces by the number and the layout of the mirror(s). For example, when two mirrors are facing each other, there could be infinitely recursive virtual image spaces, but when two mirrors are placed back against each other, there will be just one virtual image behind each mirror. To provide a guideline for the design of our module, we choose two scenes consisting of two mirrors with different layouts from our synthesized part of the dataset and train NeRF-based variants of different sub-space numbers and different feature dimensions.

We construct our variants based on NeRF with the output feature dimensions $d \in \{24, 48, 64\}$ and the number of sub-spaces $K \in \{2, 4, 6, ..., 16\}$, then we train our models on the two scenes and report the results using PSNR.

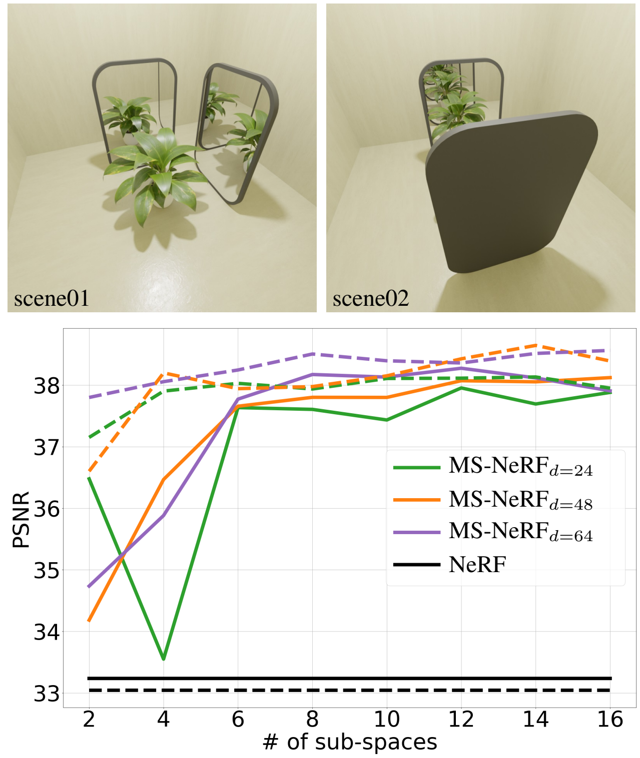


Figure 12. We use PSNR to quantitatively evaluate the ablation experiments on scene01 and scene02 and plot the results with solid and dotted lines, respectively.

Our results in Fig. 12 show that the number of sub-spaces is not required to match the actual number of virtual image spaces, and 6 sub-spaces can guarantee stable learning for multi-space radiance fields. Moreover, feature fields with dimension $d = 24$ already encode enough information for composition, but for stable performance $d = 48$ is better.

6. Conclusion

In this paper, we tackle the long-standing problem of rendering reflective surfaces in NeRF-based methods. We introduce a multi-space NeRF method that decomposes the Euclidean space into multiple virtual sub-spaces. Our proposed MS-NeRF approach achieves significantly better results compared with conventional NeRF-based methods. Moreover, a light-weighted design of the MS module allows our approach to serve as an enhancement to the conventional NeRF-backbone networks. We also constructed a novel dataset for the evaluation of similar tasks, hopefully, helping future researches in the community.

Acknowledgment: This work is funded by the Natural Science Foundation of China (NO. 62132012).

References

[1] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *ICCV*, pages 5855–5864, 2021. 2, 6

[2] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fiecvprlds. In *IEEE CVPR*, pages 5470–5479, 2022. 1, 2, 5, 6

[3] Mojtaba Bemana, Karol Myszkowski, Jeppe Revall Frisvad, Hans-Peter Seidel, and Tobias Ritschel. Eikonal fields for refractive novel-view synthesis. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–9, 2022. 2, 5

[4] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *IEEE CVPR*, pages 16123–16133, 2022. 2

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

[6] Xingyu Chen, Qi Zhang, Xiaoyu Li, Yue Chen, Ying Feng, Xuan Wang, and Jue Wang. Hallucinated neural radiance fields in the wild. In *IEEE CVPR*, pages 12943–12952, 2022. 1

[7] Zhiqin Chen, Thomas Funkhouser, Peter Hedman, and Andrea Tagliasacchi. Mobilenerf: Exploiting the polygon rasterization pipeline for efficient neural field rendering on mobile architectures. *arXiv:2208.00277*, 2022. 2

[8] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2022. 3, 5

[9] Yao Feng, Jinlong Yang, Marc Pollefeys, Michael J Black, and Timo Bolkart. Capturing and animation of body and clothing from monocular video. *arXiv:2210.01868*, 2022. 2

[10] Stephan J Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. Fastnerf: High-fidelity neural rendering at 200fps. In *ICCV*, pages 14346–14355, 2021. 2

[11] Michelle Guo, Alireza Fathi, Jiajun Wu, and Thomas Funkhouser. Object-centric neural scene rendering. *arXiv:2012.08503*, 2020. 2

[12] Yuan-Chen Guo, Di Kang, Linchao Bao, Yu He, and Song-Hai Zhang. Nerfren: Neural radiance fields with reflections. In *IEEE CVPR*, pages 18409–18418, 2022. 1, 2, 3, 5, 6

[13] Peter Hedman, Pratul P Srinivasan, Ben Mildenhall, Jonathan T Barron, and Paul Debevec. Baking neural radiance fields for real-time view synthesis. In *ICCV*, pages 5875–5884, 2021. 5

[14] Xin Huang, Qi Zhang, Ying Feng, Hongdong Li, Xuan Wang, and Qing Wang. Hdr-nerf: High dynamic range neural radiance fields. In *IEEE CVPR*, pages 18398–18408, 2022. 2

[15] Ajay Jain, Ben Mildenhall, Jonathan T Barron, Pieter Abbeel, and Ben Poole. Zero-shot text-guided object generation with dream fields. In *IEEE CVPR*, pages 867–876, 2022. 2

[16] Nishant Jain, Suryansh Kumar, and Luc Van Gool. Robustifying the multi-scale representation of neural radiance fields. *arXiv:2210.04233*, 2022. 1, 3

[17] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *IEEE CVPR*, pages 406–413, 2014. 2, 5

[18] Boyi Jiang, Yang Hong, Hujun Bao, and Juyong Zhang. Selfrecon: Self reconstruction your digital avatar from monocular video. In *IEEE CVPR*, pages 5605–5615, 2022. 2

[19] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM TOG*, 36(4):1–13, 2017. 2, 5

[20] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *ICCV*, pages 5741–5751, 2021. 1, 3

[21] 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 *IEEE CVPR*, pages 7210–7219, 2021. 1

[22] Nelson Max. Optical models for direct volume rendering. *IEEE TVCG*, 1(2):99–108, 1995. 3

[23] Ben Mildenhall, Peter Hedman, Ricardo Martin-Brualla, Pratul P Srinivasan, and Jonathan T Barron. Nerf in the dark: High dynamic range view synthesis from noisy raw images. In *IEEE CVPR*, pages 16190–16199, 2022. 2

[24] 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. *ACM TOG*, 38(4):1–14, 2019. 2, 5

[25] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 1, 2, 3, 5, 6

[26] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM TOG*, 41(4), jul 2022. 2

[27] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *IEEE CVPR*, pages 11453–11464, 2021. 2, 4

[28] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *ICCV*, pages 5589–5599, 2021. 2

[29] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *ICCV*, pages 5865–5874, 2021. 1

[30] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *IEEE CVPR*, pages 10318–10327, 2021. 1

[31] Ravi Ramamoorthi et al. Precomputation-based rendering. *Foundations and Trends® in Computer Graphics and Vision*, 3(4):281–369, 2009. 2

[32] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *ICCV*, pages 14335–14345, 2021. 2, 4

[33] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *IEEE CVPR*, pages 4104–4113, 2016. 2

[34] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *IEEE CVPR*, pages 5459–5469, 2022. 2

[35] Jiaming Sun, Xi Chen, Qianqian Wang, Zhengqi Li, Hadar Averbuch-Elor, Xiaowei Zhou, and Noah Snavely. Neural 3d reconstruction in the wild. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–9, 2022. 1

[36] Ayush Tewari, Justus Thies, Ben Mildenhall, Pratul Srinivasan, Edgar Tretschk, W Yifan, Christoph Lassner, Vincent Sitzmann, Ricardo Martin-Brualla, Stephen Lombardi, et al. Advances in neural rendering. *Computer Graphics Forum*, 41(2):703–735, 2022. 1

[37] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In *ICCV*, pages 12959–12970, 2021. 1

[38] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T. Barron, and Pratul P. Srinivasan. Ref-NeRF: Structured view-dependent appearance for neural radiance fields. *CVPR*, 2022. 2, 5, 6, 7

[39] Can Wang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. Clip-nerf: Text-and-image driven manipulation of neural radiance fields. In *IEEE CVPR*, pages 3835–3844, 2022. 2

[40] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *Advances in Neural Information Processing Systems*, 34:27171–27183, 2021. 2

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

[42] Suttisak Wizadwongsa, Pakkapon Phongthawee, Jiraphon Yenphraphai, and Supasorn Suwajanakorn. Nex: Real-time view synthesis with neural basis expansion. In *IEEE CVPR*, pages 8534–8543, 2021. 2, 5

[43] Bangbang Yang, Yinda Zhang, Yinghao Xu, Yijin Li, Han Zhou, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. Learning object-compositional neural radiance field for editable scene rendering. In *ICCV*, pages 13779–13788, 2021. 2

[44] Guo-Wei Yang, Wen-Yang Zhou, Hao-Yang Peng, Dun Liang, Tai-Jiang Mu, and Shi-Min Hu. Recursive-nerf: An efficient and dynamically growing nerf. *IEEE TVCG*, 2022. 2

[45] Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In *IEEE CVPR*, pages 1790–1799, 2020. 2, 5

[46] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. In *ICCV*, pages 5752–5761, 2021. 2

[47] Yu-Jie Yuan, Yang-Tian Sun, Yu-Kun Lai, Yuewen Ma, Rongfei Jia, and Lin Gao. Nerf-editing: geometry editing of neural radiance fields. In *IEEE CVPR*, pages 18353–18364, 2022. 2

[48] Kaan Yücer, Alexander Sorkine-Hornung, Oliver Wang, and Olga Sorkine-Hornung. Efficient 3d object segmentation from densely sampled light fields with applications to 3d reconstruction. *ACM TOG*, 35(3):1–15, 2016. 2

[49] Jason Zhang, Gengshan Yang, Shubham Tulsiani, and Deva Ramanan. Ners: Neural reflectance surfaces for sparse-view 3d reconstruction in the wild. *Advances in Neural Information Processing Systems*, 34:29835–29847, 2021. 1

[50] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv:2010.07492*, 2020. 1, 2

[51] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE CVPR*, pages 586–595, 2018. 6

[52] Yufeng Zheng, Victoria Fernández Abrevaya, Marcel C Bühler, Xu Chen, Michael J Black, and Otmar Hilliges. Im avatar: Implicit morphable head avatars from videos. In *IEEE CVPR*, pages 13545–13555, 2022. 2