SOMSI: Spherical Novel View Synthesis with Soft Occlusion Multi-Sphere Images

Tewodros Habtegebrial [*†] Christiano Gava [*†] Marcel Rogge [*†] Didier Stricker [*†] Varun Jampani [‡]
[*]TU Kaiserslautern [†]DFKI [‡]Google Research

Abstract

Spherical novel view synthesis (SNVS) is the task of estimating 360° views at dynamic novel views given a set of 360° input views. Prior arts learn multi-sphere image (MSI) representations that enable fast rendering times but are only limited to modelling low-dimensional color values. Modelling high-dimensional appearance features in MSI can result in better view synthesis, but it is not feasible to represent high-dimensional features in a large number (> 64*) of MSI spheres. We propose a novel MSI representation called Soft Occlusion MSI (SOMSI) that enables modelling high-dimensional appearance features in MSI while retaining the fast rendering times of a standard MSI. Our key insight is to model appearance features in a smaller set (*e.g. 3*) of occlusion levels instead of larger number of MSI levels. Experiments on both synthetic and real-world scenes demonstrate that using SOMSI can provide a good balance between accuracy and run-time. SOMSI can produce considerably better results compared to MSI based MODS [1], while having similar fast rendering time. SOMSI view synthesis quality is on-par with state-of-the-art NeRF [24] like model while being 2 orders of magnitude faster. For code, additional results and data, please visit* https://tedyhabtegebrial.github.io/somsi.

1. Introduction

The advent of low-cost 360° imaging devices makes spherical images a standard choice of representation for 3D scenes in comparison to more expensive 3D modelling with artists or depth sensors. Spherical images are widely used to capture and visualize 360° views of scenes with several applications in virtual tourism, navigation, advertisement, etc. However, a spherical image alone delivers a limited viewing experience with only rotations around the center. User navigation (translation) is usually enabled by capturing multiple spherical images thereby allowing the user to hop from one to another. Spherical novel view synthesis (SNVS) is the task of estimating in-between 360° views making it possible for seamless continuous user navigation in a scene. See Fig. 1 (left) for a sample illustration of the problem setting.

A practical SNVS system would have the following properties: 1. *High-quality* synthesis of disoccluded content and view-dependent effects in novel views, 2. *Fast synthesis time* enabling realtime user navigation in a scene and, 3. *Low memory* consumption to run SNVS on mobile hardware such as VR headsets. Satisfying all these properties is quite challenging. Current SNVS techniques are based on Multi-Sphere Image (MSI) representation [1, 4]. MSI can be seen as a spherical extension of Multi-Plane Images (MPI) [32, 37], which are widely adopted in view synthesis of common perspective images. More specifically, MSIs represent a scene as a set of textured spheres centered around a reference point. A key advantage of using MSIs is that rendering spheres is extremely efficient with standard rendering softwares. The simplicity of the rendering and seamless integration with graphics software makes MSIs an appealing choice for real-time rendering applications. On the other hand, current MSI based techniques such as MatryODSHkha (MODS) [1] suffer from unsatisfactory quality in synthesized novel views.

The use of Coordinate Multi-Layer Perceptrons (CMLP) is revolutionizing the field of novel view synthesis with very high-quality results such as in NeRF [24]. A key drawback of CMLP based techniques is the requirement of large number of training views as well as slow rendering. Several very recent works try to improve NeRF-like techniques in different aspects: improving rendering speed [13, 19, 26], modelling reflectance properties [3], working with in-the-wild images [23], re-lighting [30], generalizing across scenes [36], etc. Even though one could adapt a NeRF-like technique for SNVS task, rendering novel 360° views would be prohibitively slow. Several concurrent techniques to improve rendering speed of NeRFs [13, 19, 26] are either specific to perspective images or did not demonstrate their use for spherical images in the SNVS task.

In this work, we propose a novel SNVS technique that provides a good trade-off between different favourable properties: high-quality view synthesis and fast runtime with low memory requirements. Following [1], we also make use of MSI representation for fast rendering. [1]

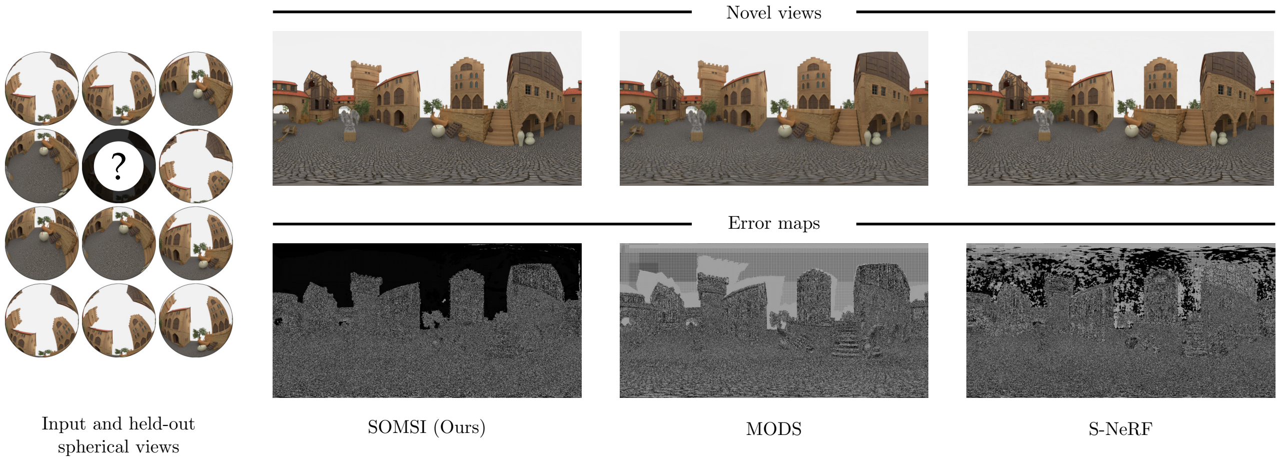

Figure 1. **High quality view synthesis with SOMSI.** (Left) Illustration of sample input and held-out target spherical views of a scene. (right) Synthesized novel views and the corresponding error maps for SOMSI (Ours), MODS [1] and S-NeRF [24].

learns a CNN that takes spherical images as input and produces an MSI representation where each point in each layered sphere has an RGB color and an alpha value associated with it. Several recent MPI works [17, 21, 34] on view synthesis demonstrate the use of high-dimensional learned features in MPI planes instead of color MPIs. Following these approaches and coordinate MLPs, we propose to learn high-dimensional appearance features in MSI using coordinate MLPs. In essence, we combine the strengths of coordinate MLPs with MSI representation, thereby achieving both high quality and fast runtime (rendering).

A key issue in representing high-dimensional features at each point in MSI is high memory consumption. For instance, representing f-dimensional features in d (usually > 64) spheres with m points each leads to a memory complexity of $\mathcal{O}(m \times d \times f)$. Since the number of points m in a 360° spherical image are usually high, the memory complexity will be prohibitively high if we represent high-dimensional features in each MSI sphere. We argue that representing dense features in MSI sphere is superfluous as much of the 3D space is empty. As a remedy, we propose a novel MSI scene representation where the scene appearance and geometry are factored into two separate data structures. We call this novel MSI representation Soft Occlusion MSI (SOMSI). SOMSI represents the scene geometry with standard MSI data structure, while the scene appearance is represented as a set of layered 2D feature maps. The key in SOMSI is to represent appearance feature maps using a smaller set of scene-specific occlusion levels instead of large set of predefined MSI spheres. In essence, SOMSI representation takes $\mathcal{O}(m \times d \times k)$ memory for scene geometry with soft occlusion masks and $\mathcal{O}(m \times k \times f)$ memory to represent scene appearance features, where k denotes the number of occlusion levels. This strategy scales much better in terms of memory with increasing feature dimension as the number of occlusion levels are significantly lower ($k = 3$ in our case) compared to spheres in MSI (usually $d > 64$). We also propose a novel SOMSI rendering formulation that allows the fast rendering of novel views like with standard MSI representation.

We demonstrate the effectiveness of our SOMSI technique with results on both synthetic and real-world scenes. Fig. 1 shows sample input and held-out spherical views along with sample results of different techniques. Results show that our approach can considerably outperform previous MSI based techniques [1]. Our approach can produce high quality novel views that are on-par with spherical adaptation of NeRF [24] technique (S-NeRF) while being 2 orders of magnitude faster, with rendering time close to MSI techniques [1]. We make the following contributions:

- We propose a novel Soft Occlusion Spherical Multi-Sphere (SOMSI) representation that can effectively scale to encode high-dimensional scene appearance features in MSI representation using learnable occlusion layers.
- We propose an efficient way to render novel views from the learned SOMSI representation.
- Our approach effectively combines the advantages of different techniques with high quality view synthesis that is on-par with implicit volumetric representations [24] while having fast runtime like with standard MSI representations [1].

2. Related Work

View synthesis research has a long history in computer vision and graphics starting from the seminal work of image-space morphing in Chen and Williams [8], light field rendering [20], Lumigraphs [5, 15] followed by multi-view stereo reconstruction techniques [6, 7, 10, 14, 18]. Here, we briefly review the relevant learning based techniques.

Novel View Synthesis. One of the earliest learning based IBR techniques is the DeepStereo method by Flynn *et al.* [12]. DeepStereo trains a CNN to produce novel views from input plane-sweep volumes. An important milestone for the learning based IBR research was the re-introduction of the Multi-plane Images (MPI) by Zhou *et al.* [37]. The view synthesis capabilities of MPIs have been pushed ever further by the DeepView [11] technique. DeepView combines the MPI scene representation with a learned gradient descent based optimization to render highly accurate novel views of challenging real world scenes. A recent extension of MPIs model view dependent effects [35]. Recently, Neural Radiance Fields (NeRF) [24] techniques uses Coordinate Multi-layer Perceptrons to model a given scene resulting in very high-quality view synthesis results. NeRF renders a scene by performing standard volumetric rendering. However, the visibility and color information for every point in the rendering volume are determined by invoking the trained MLP which is time-consuming.

Spherical View Synthesis. Single spherical images or panoramic stereo cannot provide parallax as head movement (translation) is not possible. Rendering panoramic scenes with motion parallax is studied in several works [2, 22, 29]. Synthesizing novel views "on demand" is key to enhance user experience by allowing for head movement. Broxton *et al.* [4] presented a light-weight immersive light field video rendering technique by extending MPI formulation of DeepView [11] into a Multi-sphere Image (MSI) representation. Moreover, sparse set of MSI spheres are used in order to create a light-weight layered mesh representation that can be rendered on mobile and web platforms. Concurrently with Bronxton *et al.* [4], the MODS [1] technique showed the effectiveness of MSIs for 360° view synthesis. MSIs are an attractive option for SNVS as they support real-time rendering with standard rendering softwares. In this paper we build upon the MSI scene representation and propose a novel MSI reprsentation that can more efficiently model high-dimensional appearance features.

3. Preliminaries

Spherical Image Representation. A spherical image is an environment mapping that captures the entire visible scene from a single point in space. In other words, it comprises a 360°× 180° panoramic view of the surroundings of the camera. Spherical images are usually stored as a 2D pixel map of dimensions $h \times w$ often referred to as Equirectangular Projection (ERP). Each ERP pixel $\mathbf{p} = [u, v]^T$, where $u \in [0, w-1], v \in [0, h-1]$, corresponds to a point on a unit sphere $[\theta, \phi, 1]^T$ expressed in spherical coordinates. This mapping between a point on the unit sphere and its pixel location on the ERP is given as

$$u = w\,(1 - \frac{\theta}{2\,\pi}), \qquad v = h\,\frac{\phi}{\pi}, \tag{1}$$

with x-axis pointing into the screen, y-axis to the left and z-axis up; a Cartesian point $\mathbf{x} = [x, y, z]^T$ is converted into spherical coordinates $\mathbf{x}^s = [\theta, \phi, r]^T$ as

$$\theta = atan(\frac{y}{x}) \quad \phi = acos(\frac{z}{r}) \quad r = \sqrt{x^2 + y^2 + z^2}. \tag{2}$$

A cartesian point $\mathbf{x}$ in 3D can be projected on to an ERP by first obtaining $\mathbf{x}^s = [\theta, \phi, r]^T$ with Equation 2, followed by mapping $\mathbf{x}^s$ to the ERP location $[u, v]^T$ via Equation 1.

Multi-Sphere Images (MSI). This is the most commonly used representation for SNVS [1, 4]. MSI can be seen as a spherical extension of Multi-Plane Images (MPI) that are widely used in view synthesis literature [32, 37]. An MSI is composed of a set of concentric $RGB\alpha$ spheres. The use of spheres to represent a scene allows real-time rendering and easy integration with common rendering softwares such as Unity3D [16] and Blender [9]. This makes MSIs highly suitable for downstream VR applications.

Formally, an MSI is a set of concentric spheres of radii $\{r_i\}_{i=1}^d$ with each sphere representing a spherical image. The r_i values are set by linearly sampling the inverse depth range between predefined near r_{near} and far r_{far} values. Each of the d spheres in an MSI has transparency $\alpha \in [0,1]^{m\times 1}$ and color $C \in [0, 255]^{m\times 3}$, where $m = h \times w$ is the pixel resolution of the spherical images and d is the number of spheres. The number of spheres controls the trade-off between the fidelity of the scene representation and the computational costs (rendering time and memory). Increasing d yields higher fidelity but requires more memory and results in slower rendering. These spherical images can be represented either in 3D with spherical coordinates or with 2D ERP planes.

MSI Rendering. Fig. 2 shows an illustration of MSI spheres centered around reference view $\mathbf{r}$. Suppose we want to render a spherical image from a novel view center $\mathbf{t}$, we shoot rays from the target center and alpha composite the colors along the intersection points w.r.t. reference-view MSI spheres. Specifically, for a sample ray direction (corresponding to a location $\mathbf{p} = [u, v]^T$ in the target ERP), we first compute the ray intersections $\mathbf{x}_i^p \in \mathbb{R}^3; i \in \{1, ..., d\}$ with the d MSI spheres using a standard sphere-ray intersection technique [25]. We then record MSI transparency and color values at these intersection points: $\{(\alpha_i^p, C_i^p)\}_{i=1}^d$. The final color C^p for target ray $\mathbf{p}$ is calculated by over-compositing C_i^ps with α_i^ps in a back-to-front manner (also known as alpha composition):

$$C^p = \sum_{i=1}^{d} \alpha_i^p\, C_i^p \prod_{j<i} (1 - \alpha_j^p). \tag{3}$$

The same process is used for all the ray directions in the target view to obtain final novel view spherical image at the target location. Since this process is trivially parallelizable, it is straightforward to use GPUs for fast rendering of novel views. One could also export MSI spheres as textured meshes to leverage standard rendering engines for real-time novel view rendering.

4. Approach

Problem. We tackle the problem of Unstructured Spherical Light-Field Interpolation also referred to as Spherical View Synthesis: Given a set of 360° spherical images captured at different locations in a scene, the aim is to estimate spherical images from novel camera viewpoints. Formally, the input to our method is a set of n spherical images, $I_i \in \mathbb{R}^{m \times 3}; i \in \{1, ..., n\}$ each with m pixels, and their $SE(3)$ camera poses, $P \in \mathbb{R}^{n \times 3 \times 4}$. Given a set of posed spherical images, we learn a scene representation from which one can dynamically render novel views from target camera poses.

Fig. 3 illustrates the overview of our spherical novel view synthesis (SNVS) technique. We consider one of the input camera poses as reference and optimize a coordinate MLP network that learns to estimate a novel scene representation called 'Soft Occlusion MSI' (SOMSI) at that reference location. We can then use this representation to dynamically render novel views from this scene representation at the target locations. Next, we describe the SOMSI representation and how we can render novel views from that representation.

4.1. Soft Occlusion MSI (SOMSI)

Motivation. Even though standard MSI representation is efficient in terms of rendering speed, it is limited to low-dimensional RGB appearance in each MSI sphere. Recent works on view synthesis [17, 21, 34] show that it is beneficial to represent the appearance in 3D scene with high-dimensional deep features instead of simple RGB colors. Representing higher dimensional features at each MSI sphere is memory intensive as representing f-dimensional features at each point in each MSI sphere leads to a memory complexity of $\mathcal{O}(m \times d \times f)$ which is not feasible even for moderate values of $f > 10$.

In this work, we propose a novel MSI representation called *Soft occlusion MSI* (SOMSI) which scales much better with increasing appearance feature dimensionality. The key to our technique is decoupling appearance features from scene geometry in MSI representation. In SOMSI, we use standard MSI ERPs to represent geometry and soft occlusions; and use a small set of occlusion layers/ERPs to represent appearance features. Our key insight is that much of the points in MSI spheres are empty and we can represent multi-sphere appearance features with a much smaller set of occlusion appearance features.

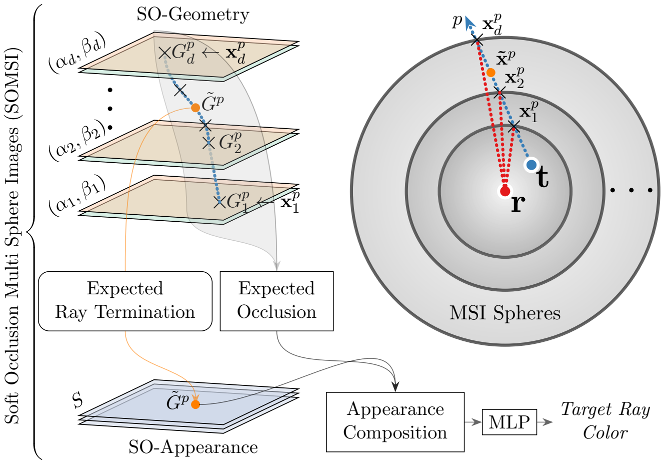

Figure 2. **SOMSI Rendering.** Illustration of SO-Geometry and SO-Appearance planens in our SOMSI scene representation in the reference view at $\mathbf{r}$. Rendering a target ray p involves computing expected target ray termination and expection occlusion level, using which we can composite the appearance features to estimate the final target ray color.

SOMSI Representation. As illustrated in Fig. 2, SOMSI representation has two sets of ERPs: *SO-Geometry* and *SO-Appearance*. SO-Appearance comprises of soft occlusion appearance features $S \in \mathbb{R}^{m \times k \times f}$, where f denotes the size of color/appearance descriptors and k the number of occlusion layers. Each of the k ERPs denotes the scene appearance at a specific occlusion layer. The first ERP represents all the visible surface appearances in the reference view; second ERP represents the appearance of occluded surfaces that are behind the visible surfaces and the third ERP represents the further occluded surfaces and so on. In practice, we observe that 3 layers ($k = 3$) are enough to represent most occluded content in general scenes. In short, SO-Appearance represents the scene appearance features with much smaller set ($k = 3$) of occlusion layers in comparison to larger number ($d > 32$) of ERPs in a standard MSI representation.

SO-Geometry, on other hand, represents scene geometry using all the d spheres. Specifically, SO-Geometry consists of multi-sphere transparencies $\alpha \in \mathbb{R}^{m \times d}$ and soft occlusion masks $\beta \in \mathbb{R}^{m \times d \times k}$. As the name indicates, transparencies α represent the sphere transparencies like in standard MSI representation. The k-dimensional soft-occlusion mask β_i^p of a 3D point at i^{th} sphere along ray/pixel [1] p represents the occlusion level of that 3D point $\mathbf{x}_i^p$ in a soft manner. For example, β_i^p $[0, 1, 0]$ denotes that $\mathbf{x}_i^p$ belongs

[1] Note: We often refer to pixels as rays since ERP pixels correspond to a specific ray direction.

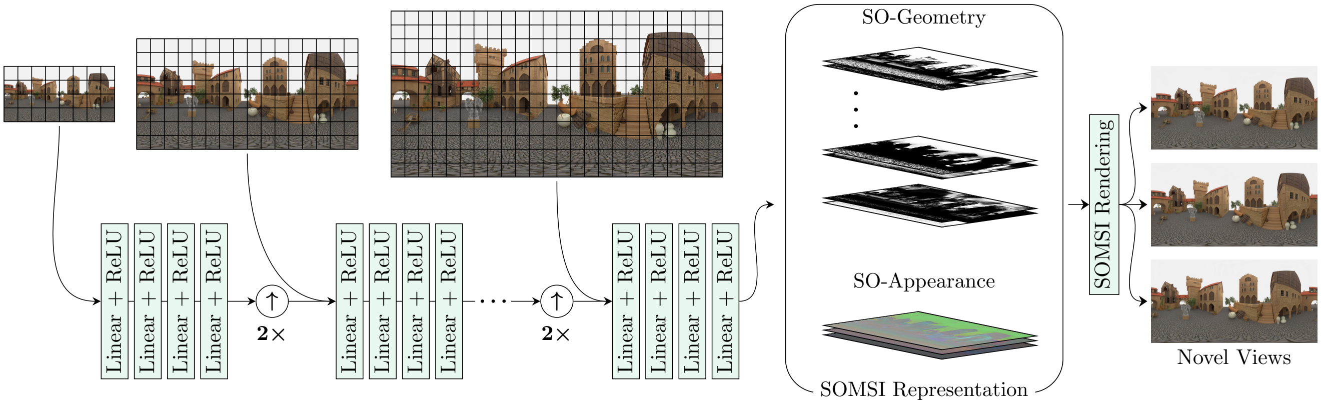

Figure 3. **Approach Overview.** We learn an MLP that takes spherical locations and colors from multi-resolution reference view images; and estimates SOMSI scene representation at those locations. We can then efficiently render novel view images from the learned SOMSI.

to second occlusion level i.e., occluded by one visible surface. The soft occlusion masks β provide the association of planes in SO-Geometry and SO-Appearance. SO-Geometry and SO-Appearance planes have the memory complexity of $\mathcal{O}(m \times d \times k)$ and $\mathcal{O}(m \times k \times f)$ respectively. Since $k << d$, the combined memory complexity of these two will be smaller than representing f-dimensional features at each of the d planes, which has $\mathcal{O}(m \times d \times f)$ memory complexity. This makes SOMSI representation scale better with higher dimensional features while retaining the fast rendering times of MSI representation.

SOMSI Rendering. Novel view rendering with SOMSI is not a simple alpha composition process, (Eqn. 3) as we do not represent appearance features at each of the MSI planes. One can simply convert SO-Appearance features into full MSI planes, and then can use alpha composition for novel view rendering. Converting SO-Appearance features into full MSI feature planes would again incur $\mathcal{O}(m \times d \times f)$ memory complexity which we want to avoid. In contrast, we propose a technique that directly utilizes SO-Appearance features for rendering. Our SOMSI rendering along each ray from the target center has three main steps: 1. Computing expected ray termination on the target ray; 2. expected occlusion along the target ray; and then 3. using both these estimates to composite novel view appearance features from SO-Appearance layers. Let us discuss each of these steps in detail.

1. Expected Ray Termination. Following the same notation as in Sec. 3 and as illustrated in Fig. 2, let us suppose we learn an SOMSI centered at reference $\mathbf{r}$ and we want to do the rendering from target location $\mathbf{t}$. Consider the rendering of target appearance features along the sample target ray direction p, as shown in Fig. 2. Similar to standard MSI rendering, we first compute these target ray intersections $\mathbf{x}_i^p \in \mathbb{R}^2; i \in \{1, ..., d\}$ with the d MSI spheres (in reference view) using standard sphere-ray intersection technique [25]. Each of these points $\mathbf{x}_i^p$ has the corresponding 2D points in SO-Geometry ERP planes with locations $G_i^p \in \mathbb{R}^2$, transparency α_i^p and soft occlusion β_i^p as shown in Fig. 2. These points $\{G_i^p\}_{i=1}^d$ lie on an epipolar line in the reference view. We compute the 2D ERP location (of ray p in the reference camera) $\tilde{G}^p \in \mathbb{R}^2$ of the target ray p as:

$$\tilde{G}^p = \sum_{i=1}^{d} \alpha_i^p \, G_i^p \prod_{j<i} (1 - \alpha_j^p), \tag{4}$$

which is an alpha composition of 2D ERP locations $\{G_i^p\}_{i=1}^d$ instead of color values as in standard MSI rendering (Eqn. 3). The expected ray on the Epipolar line, $\tilde{G}^p$ is a 2D location in SOMSI. This point effectively computes the location of target ray termination in the reference camera (denoted as orange dots in Fig. 2). Note that, one can equivalently compute the ray termination in $3D$ by alpha compositing the ray-sphere intersection points and project the result to the reference ERP to get $\tilde{G}^p$.

2. Expected Occlusion. The expected ray termination $\tilde{G}^p$ at the reference view provides the correspondence, in the reference view, for the target ray p. Thus, we can use this ERP location $\tilde{G}^p$ to get the corresponding appearance features from SO-Appearance. Since SO-Appearance has different occlusion layers, we also need to estimate expected occlusion level for the target ray p. We compute the expected occlusion level $\tilde{S}^p \in \mathbb{R}^k$ using the same alpha compositing formulation, but compositing the soft-occlusion masks β in back-to-front manner:

$$\tilde{\beta}^p = \sum_{i=1}^{d} \alpha_i^p \, \beta_i^p \prod_{j<i} (1 - \alpha_j^p). \tag{5}$$

3. Appearance Composition. To compute an appearance feature for the target ray/pixel p, we first bilinearly interpolate the SO-Appearance features S at the nearest grid points resulting in a multi-layer appearance feature $\tilde{S}^p \in \mathbb{R}^{k \times f}$. We then multiply this feature map with the estimated soft occlusion mask $\tilde{\beta}^p$ to obtain a single appearance feature for the target ray: $\tilde{Q}^p = \tilde{S}^p \tilde{\beta}^p$. Multiplying $\tilde{\beta}^p$ with $\tilde{S}^p$ effectively chooses the SO-Appearance feature $\tilde{Q}^p \in \mathbb{R}^f$ at the expected occlusion level in a soft manner. In summary, we render appearance features along a target ray by first computing the corresponding ERP location in the reference view and then choosing the appearance feature at that ERP location and the expected occlusion level.

Synthesizing Novel View ERPs. We still need to convert the appearance feature $\tilde{Q}^p$ into RGB color for the target ray p. We use a simple MLP to convert $\tilde{Q}^p$ into target ray color $\tilde{C}^p \in \mathbb{R}^3$. We repeat the same SOMSI rendering processing for all the target rays followed by conversion to color values to obtain the complete novel view spherical image.

Modeling view dependent effects. Color based MSI approaches such as MODS [1] can not robustly model view dependent effects such as specular reflections in target views as color values are baked in the reference MSI. Following the neural basis decomposition approach [35], our SO-Appearance model can also incorporate view dependent effects in a natural way. A neural basis decomposition approach replaces each RGB color value (in the MPI or MSI) by reflectance coefficients ($\omega \in \mathbb{R}^{e \times 3}$, where e is the number of reflection coefficients). The coefficient images can be combined into a single color image using learned basis functions $\{\gamma_i : \mathbb{R}^3 \mapsto \mathbb{R}^1\}_{i=1}^{e-1}$, as follows $C = \omega_0 + \sum_{i=1}^{e-1} \gamma_i \omega_i$. γ_i are the scalar outputs of MLP networks that take viewing direction as input.

Applying this view dependent modelling scheme directly in a standard MSI representation leads to a high memory complexity of $\mathcal{O}(m \times d \times 3 \times e)$ as we need to represent e-dimensional reflectance coefficients for each color value at each MSI point. For MPIs, to overcome this issue, NeXMPI [35] proposes a *coefficient sharing* strategy by grouping different MPI layers and using same reflectance coefficients within each MPI layer group. Eventhough one could use similar grouping strategy in MSI representation, the grouping of MSI spheres is somewhat arbitrary. Our SO-Appearance layers provides a more physically meaningful grouping of MSI spheres based on occlusions and thus provide a natural choice to efficiently represent reflectance coefficients with $\mathcal{O}(m \times k \times 3 \times e)$ instead of $\mathcal{O}(m \times d \times 3 \times e)$. More specifically, just like appearance features, we need to represent reflectance coefficients only at k occlusion layers in SO-Appearance instead of much larger d layers in MSI. In the experiments, we observe that using modelling view dependent effects with reflectance coefficients in SO-Appearance improves the view synthesis results (Table 4).

4.2. Learning SOMSI with Coordinate MLP

We learn a scene-specific SOMSI from a given set of input spherical images captured at different 3D locations in a scene. Following the recent success of coordinate MLPs for view synthesis [23,24], we make use of MLPs that takes 2D ERP pixel locations as input and produces SOMSI scene representation at those locations w.r.t. a reference view. As illustrated in Fig. 3, MLP takes the spherical coordinates $[\theta, \phi]$ for a point r along with reference image color I_{ref}^r at that location as inputs, and produces SOMSI representation at that location:

$$\mathcal{N} : (\zeta(\theta, \phi), I_{ref}^r) \mapsto (\alpha^r, \beta^r, S^r), \tag{6}$$

where $\zeta(\theta, \phi)$ denotes the Fourier embedding of the coordinates as used in recent coordinate networks such as [24]. Contrasting to existing works that only feed Fourier embedded coordinates into MLPs, we also input RGB color values I_{ref} as well to the MLP. We observe faster training convergence and better novel view synthesis quality with additional I_{ref} as input to the network. Another distinction in our network is the use of multi-resolution coordinate maps. As shown in Fig. 3, we start with low-resolution grid as network input and upsample the grid locations as well as input spherical image after every few linear+ReLu layers. We again observe that this multi-resolution strategy resulted in better results compared to using a single resolution grid. Once we learn an SOMSI for a given scene, we can dynamically render spherical images from arbitrary novel view locations using the SOMSI rendering technique described in the previous section.

Training. We train the network $\mathcal{N}$ on a specific scene by randomly sampling a mini-batch of input cameras and rendering their corresponding held-out novel views. Our rendering pipeline is fully differentiable (due to our soft occlusion formulation) and thus allows for training with backpropagation. The network parameters are learned by minimizing the L_2 distance between the predicted novel view $\hat{I}^t$ and its respective groundtruth ERP image I^t for every camera in the mini-batch.

5. Experiments

We analyzed our SOMSI technique for spherical novel view synthesis on both synthetic and real-world datasets; with both structured (input views captured with cameras placed on a regular grid) and unstructured spherical light fields.

Baselines. We compare SOMSI with two baseline techniques: MODS [1] and S-NeRF [24]. MODS is a spherical view synthesis method that renders novel views from an Omnidirectional Stereo (ODS) input pair. We adapted MODS to work with a pair of ERPs as input, like in our setting. Given a spherical light field, we train MODS by randomly sampling a triplet of spherical images where the

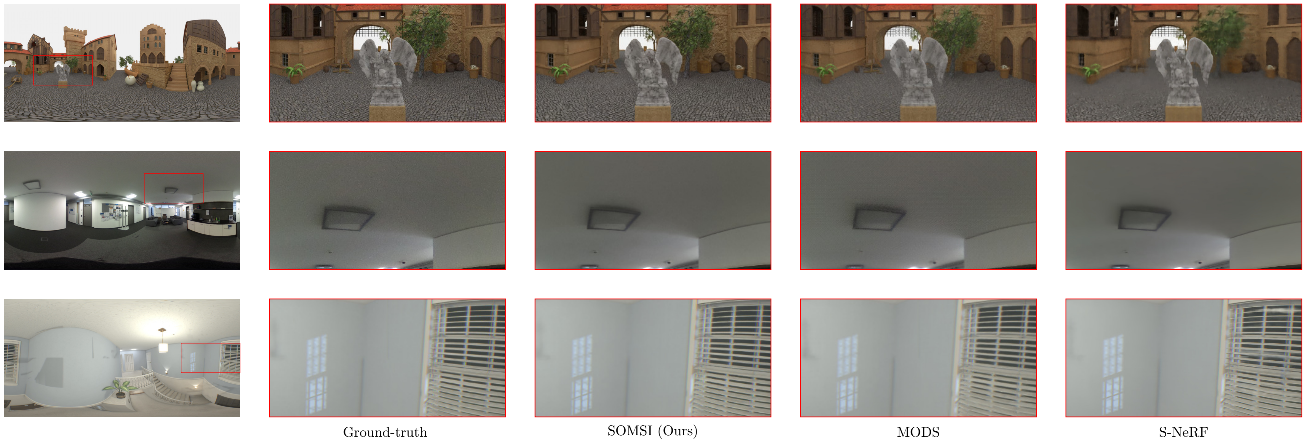

Figure 4. **Sample visual results.** Novel views on the *Sea Port* (row-1), *Coffee Area* (row-2) and *Residential* (row-3) datasets. Coffee Area is a real dataset captured with Ricoh-Theta-S camera, while the rest are synthetic. In the first row, MODS produces blurred images and struggles with thin structures. S-NeRF creates images with even less details than MODS (see the cobblestones) and far away objects (eg. the tree beyond the gate) are not well reconstructed. Specular reflections (2nd row) and thin structures (3rd row) are better captured with our method. For a better visualization, we encourage the reader to zoom-in on the insets.

	SOMSI (Ours) 13 ms		MODS [1] 10 ms		S-NeRF [24] 2710 ms	
Scene	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑
Residential	37.01	0.946	33.37	0.907	36.29	0.944
Replica	39.54	0.986	35.91	0.970	42.51	0.992
Coffee Area 1,2	32.48	0.872	28.73	0.794	32.59	0.874
Sea Port	27.67	0.825	23.79	0.626	27.76	0.825

Table 1. **Comparison with state-of-the-art.** SOMSI (Ours) can produce considerably higher quality results compared to MODS [1] with similar runtime. SOMSI produces on-par results with S-NeRF [24] while being 2 orders of magnitude faster.

first two serve as the input and the last one is used as target. During test, we render target novel views by using the first and last training cameras as input. We omitted the inverse-transform regularization in MODS [1] baseline as the MODS work itself noted that this adversely effects the view synthesis results. Our second baseline is a NeRF [24] based technique. The original NeRF technique is designed to work with perspective images. We trained NeRF with spherical images by changing the camera model from pinhole (perspective) to spherical. We refer to this spherical NeRF baseline as 'S-NeRF' in the experiments. We train all three techniques (MODS, S-NeRF and SOMSI (Ours)) on a per-scene basis.

Datsets. We experiment with a total of 20 scenes from the following 4 datasets:

- *Replica* [31] public dataset contains 18 different photo-realistic reconstructions of indoor scenes. We used 12 randomly selected scenes. We used the HabitatAPI [27] to render a 5×5 grid of spherical images, where cameras are placed 20 cm apart.
- *Residential* dataset consists of 3 synthetic scenes depicting residential houses that were created using Unity3D [16]. We create 3×3 views per scene with cameras that are 20 cm.
- *Coffee Area* dataset consists of 4 real-world scenes that are self-captured using Ricoh-Theta-S spherical camera. The first 2 of scenes have mainly diffuse objects, while the last 2 contain reflective objects. We use COLMAP [28] to calibrate the cameras. See details in the supplementary material.
- *Sea Port* dataset is an synthetic 3D model of a medieval sea port. This dataset is created by improving assets that where borrowed from blendwap. This dataset consists of a 5×5 spherical light-field with $20cm$ camera baseline.

Training and evaluation. Following MODS [1], we use 320×640 resolution spherical images for training and evaluation. Training our model takes about 18 hours on RTX-2080Ti GPU. SOMSI can be trained to up to 1024×2048 resolution images on a single RTX-A100 GPU. See the supplementary material for more details. For quantitative evaluation, we compare the predicted novel views with the held-out ground-truth views and then compute the commonly used metrics in view synthesis literature: Peak Signal-to-Noise Ratio (PSNR) and Structured Similarity Index (SSIM) [33]. Unless specified otherwise, we train our main model with: number of spheres $d = 64$, occlusion layers $k = 3$ and the appearance features size $f = 24$.

Comparisons. Table 1 shows the comparison of different metrics across different datasets and techniques. As the PSNR and SSIM metrics clearly demonstrate, SOMSI has considerably and consistently better PSNR and SSIM across all the datasets when compared to MODS [1]. SOMSI has on-par metrics compared to the spherical version of NeRF (S-NeRF) [24]. S-NeRF is a strong baseline that stores the scene using MLPs in an implicit manner. As a result, the rendering speed is quite slow (2710ms) compared to SOMSI (13ms). Both SOMSI and MODS learn explicit scene representation enabling fast rendering. In summary, SOMSI provides a good balance between rendering speed and view synthesis quality. That is, SOMSI can produce on-par view synthesis results with S-NeRF while being two orders of magnitude faster (similar rendering time as MODS).

Ablations. We perform ablations with the appearance feature dimensions f and the number of occlusion layers k in SO-Appearance features. Table 2 shows metrics with different feature dimensions $f = 3, 12, 24$ on three scenes from Replica, Residential and Sea Port. Results clearly show that both PSNR and SSIM improves with increasing feature dimensions. We observe minimal performance improvements with even higher feature dimensions $f > 24$. This observation is inline with recent view synthesis studies [17, 34] on perspective images. These results demonstrates the use of high dimensional features to model appearance details in an MSI instead of commonly used RGB values thereby justifying the need for our efficient SOMSI representation. Using SOMSI representation makes it feasible to increase the appearance feature dimensions without incurring significant memory or runtime costs, compared to representing appearance features using standard MSI representation.

Table 3 shows the view synthesis metrics with the different number of occlusion levels k. We notice some increase in performance with using 3 levels compared to 2 levels. Using more occlusion levels did not show any considerable improvement in performance. This shows that 3 occlusions levels are usually enough to represent a scene in SOMSI representation.

	f=3		f=12		f=24	
Scene	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑
Residential	36.96	0.943	36.82	0.944	**37.01**	**0.946**
Replica	38.75	0.980	38.97	0.981	**40.23**	**0.986**
CoffeArea 1,2	32.33	**0.873**	32.44	**0.873**	**32.48**	0.872
Sea-Port	27.27	0.802	27.55	0.822	**27.67**	**0.825**

Table 2. **Ablation of appearance feature dimensionality f.** Metrics show that higher feature dimensions results in better performance thereby justifying the need for our SOMSI technique that can effectively represent higher-dimensional appearance features with occlusion layers instead of a full set of MSI layers.

View Dependent Effects. For the two scenes in Coffee Area that has specular objects, we experiment with learning reflectance coefficients in SO-Appearance as described at the end of Section 4.1. Table 4 shows the PSNR and SSIM metrics on the two Coffe Area scenes with different number of reflectance coefficients. Results clearly show that the view synthesis quality improves with more reflectance coefficients. Eventhough this is not a surprising result, this further demonstrates the use of learning higher dimensional features in MSI (either appearance features or reflectance coefficients or both) compared to only RGB color values.

Limitations. One of the limitations of our technique is that the scene representation network is optimized independently for different scenes. This assumes a sufficient number of training views for each scene which may not be available for some scenes in practice. A more practical approach would be to learn a network prior that works across different scenes. A main challenge in learning such as dataset prior is that there exists no large scale spherical image dataset with diverse scenes to learn meaningful priors.

Societal Impact. Given the advent of low-cost spherical imaging and VR devices, we envision that our SNVS technique would be useful for several real-world applications such as virtual tourism. Since we train our network per scene, our approach would be less prone to dataset bias compared to networks that are learned on large-scale datasets.

	$k = 2$		$k = 3$		$k = 5$	
Scene	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑
Sea-Port	27.30	0.809	27.32	0.813	27.44	0.814
Replica	36.51	0.977	37.13	0.979	36.78	0.981

Table 3. **Ablation of occlusion levels k.** Metrics show that 3 occlusion levels are good enough to represent the scene and to obtain high-quality view synthesis results. Due to limited compute resources, here we used only the first 3 scenes from Replica.

	$e = 1$		$e = 4$		$e = 6$	
Scene	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑
CoffeArea-3	33.66	0.891	33.70	0.892	**35.86**	**0.902**
CoffeArea-4	30.49	0.865	30.53	0.867	**30.61**	**0.869**

Table 4. **The effect of number of reflectance coefficients e for view dependent effects.** Metrics demonstrate the use of higher number of refectance coefficients further emphasizing the need for modelling higher dimensional features in MSI representation.

6. Conclusion

In this work, we propose a novel multi-sphere representation called SOMSI that can effectively model high-dimensional appearance features in MSI representation that are commonly used for spherical novel view synthesis. The key is to represent features in occlusion layers instead of full set of MSI spheres. We presented a novel SOMSI rendering scheme that retains the fast rendering of standard MSI representation while producing high quality view synthesis. SOMSI also produces on-par results with NeRF technique, while being 2 orders of magnitude faster.

Acknowledgment

We thank Koretaka Ogata and Soichiro Yokota, for preparing the Residential Area scenes.

References

[1] Benjamin Attal, Selena Ling, Aaron Gokaslan, Christian Richardt, and James Tompkin. Matryodshka: Real-time 6dof video view synthesis using multi-sphere images. In *European Conference on Computer Vision (ECCV)*, pages 441–459, 2020. 1, 2, 3, 6, 7

[2] Tobias Bertel, Neill DF Campbell, and Christian Richardt. Megaparallax: Casual 360° panoramas with motion parallax. *IEEE transactions on visualization and computer graphics*, 25(5):1828–1835, 2019. 3

[3] Mark Boss, Raphael Braun, Varun Jampani, Jonathan T Barron, Ce Liu, and Hendrik Lensch. Nerd: Neural reflectance decomposition from image collections. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12684–12694, 2021. 1

[4] Michael Broxton, John Flynn, Ryan Overbeck, Daniel Erickson, Peter Hedman, Matthew Duvall, Jason Dourgarian, Jay Busch, Matt Whalen, and Paul Debevec. Immersive light field video with a layered mesh representation. *ACM Transactions on Graphics*, 39(4):86–1, 2020. 1, 3

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

[6] Gaurav Chaurasia, Sylvain Duchêne, Olga Sorkine-Hornung, and George Drettakis. Depth synthesis and local warps for plausible image-based navigation. In *ACM Transactions on Graphics*, 2013. 3

[7] Gaurav Chaurasia, Olga Sorkine, and George Drettakis. Silhouette-aware warping for image-based rendering. In *Eurographics Symposium on Rendering*, 2011. 3

[8] Shenchang Eric Chen and Lance Williams. View interpolation for image synthesis. In *ACM SIGGRAPH*, 1993. 2

[9] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. 3

[10] Martin Eisemann, Bert De Decker, Marcus Magnor, Philippe Bekaert, Edilson De Aguiar, Naveed Ahmed, Christian Theobalt, and Anita Sellent. Floating textures. In *Computer graphics forum*, volume 27, pages 409–418. Wiley Online Library, 2008. 3

[11] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3

[12] John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. Deepstereo: Learning to predict new views from the world's imagery. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5515–5524, 2016. 3

[13] Stephan J Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. Fastnerf: High-fidelity neural rendering at 200fps. *arXiv Preprint*, 2021. 1

[14] Michael Goesele, Jens Ackermann, Simon Fuhrmann, Carsten Haubold, Ronny Klowsky, Drew Steedly, and Richard Szeliski. Ambient point clouds for view interpolation. In *ACM SIGGRAPH 2010 papers*, pages 1–6. 2010. 3

[15] Steven J Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F Cohen. The lumigraph. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 43–54, 1996. 2

[16] John K Haas. A history of the unity game engine. 2014. 3, 7

[17] Tewodros Habtegebrial, Varun Jampani, Orazio Gallo, and Didier Stricker. Generative view synthesis: From single-view semantics to novel-view images. *Advances in Neural Information Processing Systems (NIPS)*, 2020. 2, 4, 8

[18] Peter Hedman, Tobias Ritschel, George Drettakis, and Gabriel Brostow. Scalable inside-out image-based rendering. *ACM Transactions on Graphics (TOG)*, 35(6):1–11, 2016. 3

[19] Peter Hedman, Pratul P Srinivasan, Ben Mildenhall, Jonathan T Barron, and Paul Debevec. Baking neural radiance fields for real-time view synthesis. *arXiv Preprint*, 2021. 1

[20] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 31–42, 1996. 2

[21] Zhengqi Li, Wenqi Xian, Abe Davis, and Noah Snavely. Crowdsampling the plenoptic function. In *European Conference on Computer Vision (ECCV)*, pages 178–196. Springer, 2020. 2, 4

[22] Bicheng Luo, Feng Xu, Christian Richardt, and Jun-Hai Yong. Parallax360: Stereoscopic 360 scene representation for head-motion parallax. *IEEE transactions on Visualization and Computer Graphics*, 24(4):1545–1553, 2018. 3

[23] 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7210–7219, 2021. 1, 6

[24] 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 *European Conference on Computer Vision (ECCV)*, pages 405–421. Springer, 2020. 1, 2, 3, 6, 7

[25] Matt Pharr, Wenzel Jakob, and Greg Humphreys. *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition, 2016. 3, 5

[26] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. *arXiv preprint arXiv:2103.13744*, 2021. 1

[27] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *IEEE International Conference on Computer Vision (ICCV)*, pages 9339–9347, 2019. 7

[28] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 7

[29] Ana Serrano, Incheol Kim, Zhili Chen, Stephen DiVerdi, Diego Gutierrez, Aaron Hertzmann, and Belen Masia. Motion parallax for 360 rgbd video. *IEEE Transactions on Visualization and Computer Graphics*, 2019. 3

[30] Pratul P Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T Barron. Nerv: Neural reflectance and visibility fields for relighting and view synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7495–7504, 2021. 1

[31] 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. *arXiv preprint arXiv:1906.05797*, 2019. 7

[32] Richard Szeliski and Polina Golland. Stereo matching with transparency and matting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1998. 1, 3

[33] 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. 7

[34] Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. SynSin: End-to-end view synthesis from a single image. In *arXiv Preprint*, 2020. 2, 4, 8

[35] Suttisak Wizadwongsa, Pakkapon Phongthawee, Jiraphon Yenphraphai, and Supasorn Suwajanakorn. Nex: Real-time view synthesis with neural basis expansion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8534–8543, 2021. 3, 6

[36] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4578–4587, 2021. 1

[37] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo Magnification: Learning view synthesis using multiplane images. In *ACM Transactions on Graphics (SIGGRAPH)*, 2018. 1, 3