Plenoptic Tunnels: Exploring Casual 360° Captures in VR

ANONYMOUS AUTHOR(S)
SUBMISSION ID: 226

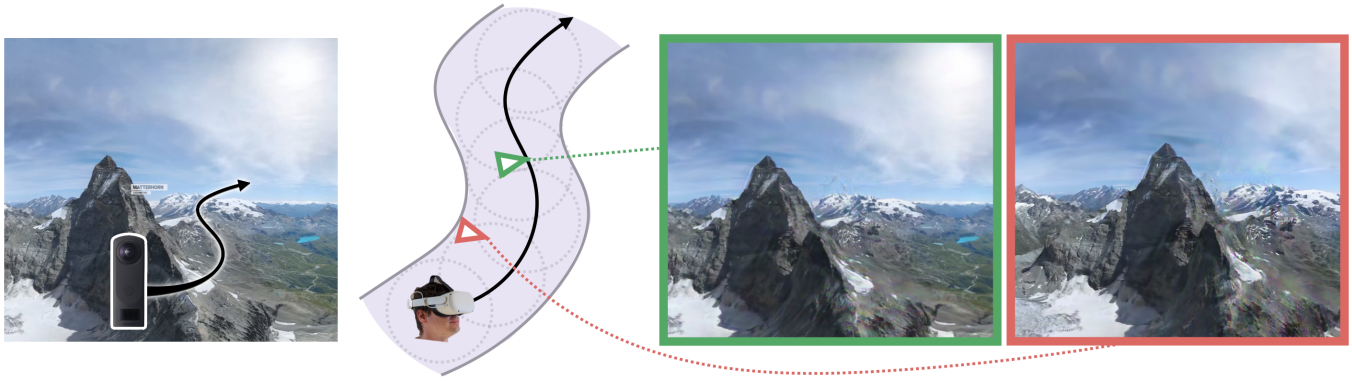


Fig. 1. From an arbitrary 360° video moving through a static scene (left), we recover a *Plenoptic Tunnel* (middle), a series of overlapping local volumetric 3D representations (right) that enable 6 DoF virtual reality exploration along the camera trajectory. Plenoptic Tunnels are optimized for rendering images at **4K** resolution, in **real-time**, on **untethered** mobile VR devices without high-end GPUs. Our approach does not require a custom rig or a prescribed capture method and thus can be applied on existing videos on the Internet, as shown here (credit: AirPano VR).

Virtual reality headsets allow us to explore captured real-world scenes with incredible immersion, but existing methods for capturing VR content are still inconvenient or inaccessible for ordinary consumers: they either require expensive and sophisticated camera rigs or deliberate capture procedures. In this paper, we present an end-to-end solution for converting a single 360° video into an immersive 6DoF experience that can be viewed on commodity devices such as untethered VR headsets and mobile phones. Our method takes as input a 360° video moving around a static scene and produces a *plenoptic tunnel*, a 6DoF explorable light field centered along the captured video trajectory. This representation is optimized for on-device rendering, producing 4K frames at >90Hz without a high-end GPU. Since our system does not prescribe a specific capture procedure, it can even be applied to previously captured videos found online, enabling immersive virtual exploration of casually captured content. We validate our system on a diverse collection of videos and provide an interactive demo that can be rendered interactively on a web browser or in VR on the Oculus Quest 2.

CCS Concepts: • **Computing methodologies** → **Virtual reality**; *Reconstruction.*

Additional Key Words and Phrases: Novel View Synthesis, Virtual Reality, Rendering, 3D Reconstruction, Light Field

ACM Reference Format:
Anonymous Author(s). 2022. Plenoptic Tunnels: Exploring Casual 360° Captures in VR. *ACM Trans. Graph.* 1, 1 (May 2022), 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

0730-0301/2022/5-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Low-cost, untethered virtual reality (VR) devices like the Oculus Quest have brought VR technology to the mainstream consumer market. Nevertheless, a large remaining bottleneck for VR is content, in particular, the ability for consumers to easily capture or generate VR experiences. Existing methods for capturing immersive real-world VR content either rely on expensive and complex capture rigs [Anderson et al. 2016; Broxton et al. 2020; Luo et al. 2018; Overbeck et al. 2018; Pozo et al. 2019], which are not available to ordinary users, or require prescribed active capture paradigms [Bertel et al. 2019, 2020; Hedman et al. 2017; Hedman and Kopf 2018], which are not applicable to previously captured videos.

In this paper, we propose an end-to-end system for converting a video into an immersive virtual reality experience viewable in real-time on an untethered VR headsets. Specifically, our method takes as input a single 360° video sequence moving around a static scene and produces a 6DoF virtual reality experience along the camera trajectory. The entire experience can be rendered with both head-motion and binocular parallax *on commodity mobile VR devices*, in high resolution (4K), and at interactive frame-rates (>90Hz). Since all we require is a 360° video, our approach can be appplied to existing videos on the Internet, as shown in Figure 1.

Our approach builds on recent progress in volumetric radiance fields for novel-view synthesis [Fridovich-Keil and Yu et al. 2022; Mildenhall et al. 2020; Wizadwongsa et al. 2021]. These methods can synthesize photorealistic novel views with view-dependent effects and capture complex scenes that are challenging to reconstruct and represent with traditional mesh representations [Hedman et al. 2017; Huang et al. 2017; Schönberger and Frahm 2016]. However, their quality comes with high computational cost, and despite the recent progress in efficiently rendering these representations [Garbin et al. 2021; Hedman et al. 2021; Reiser et al. 2021; Yu et al. 2021], these

approaches still do not meet the strict computational constraints of on-device rendering.

We address these challenges by proposing a memory and compute efficient volumetric representation for omnidirectional novel view synthesis, which we call a *Plenoptic Tunnel*. A plenoptic tunnel is a series of overlapping view-dependent, compact multi-sphere images (MSIs) that represent the local 3D structure and appearance of a scene. We present a system that solves for a plenoptic tunnel from a single input 360° video. The resulting collection of MSIs can be rendered seamlessly as a single explorable VR scene at interactive frame-rates in VR. This system includes a number of crucial optimizations such as a carefully designed factorization scheme that addresses the unique compute and memory needs of mobile VR devices. The overview of our system is illustrated in Figure 2.

We demonstrate our system on eight 360° videos depicting diverse scenes including offices, homes, and natural scenes as well as four pre-captured 360° videos collected from the Internet. We also demonstrate our approach on the OmniPhoto dataset [Bertel et al. 2020]. We ablate our system to show that the design choices made in this paper achieve comparable novel-view synthesis quality to larger, less efficient scene representations.

In summary, we present an end-to-end system that enables virtual exploration of a 360° video of a static scene on mobile and VR headset devices without requiring high-end GPUs. Our system recovers an efficient 3D scene representation that enables users to explore the scene along the camera trajectory with binocular and head motion parallax. Our system is not dependent on a specialized multi-view rig or active capture method. Note that our approach is not limited to 360° videos and can handle arbitrary image or video collections of a static scene. However, in the interest of practicality, in this paper we focus on 360° videos, since they require significantly fewer frames for full VR coverage. Upon publication, we will release our code and data, along with a web browser demo accessible by the Oculus Quest 2 and other mobile VR devices (*e.g.*, Google Cardboard).

2 RELATED WORK

A fundamental problem in virtual reality capture is synthesizing images of the scene from novel viewpoints. Any capture of a scene (e.g., an image collection or frames from a video) will always consist of a limited collection of viewpoints (i.e., a subset of the information needed to construct the scene's light field), but in order to allow the user to move their head throughout space, the rendering process must be able to produce images from *novel* viewpoints.

3D Geometry Reconstruction. Traditionally, images from novel viewpoints were created by first estimating the structure and appearance of the scene with classical 3D reconstruction techniques [Agarwal et al. 2011; Pollefeys et al. 2004; Schönberger and Frahm 2016; Sumikura et al. 2019] , involving multiple stages such as structure-from-motion (SfM) [Schönberger and Frahm 2016; Ullman 1979; Wu et al. 2011] and multi-view stereo (MVS) [Furukawa and Hernández 2015; Kutulakos and Seitz 2000; Schönberger et al. 2016]. The output of these techniques is a (sometimes textured) triangle-mesh, which can be rasterized from novel viewpoints interactively through standard graphics pipelines. Additionally, view dependent effects such as specularity and reflection can be modeled through variable projective texturing [Buehler et al. 2001; Hedman et al. 2016; Park et al. 2020; Wood et al. 2000]. While these approaches produce efficient representations for rendering, they also have significant drawbacks: geometry estimation often fails for many simple structures, such as untextured walls, thin objects, and transparent surfaces. When used in virtual reality applications, these failure modes are particularly noticeable, and can largely break the intended illusion of reality.

Volumetric novel view synthesis. Instead of explicitly reconstructing surface geometry, one can use other techniques for generating the light-rays that might have been observed from a given virtual camera viewpoint. Recent advancements in neural volumetric rendering [Barron et al. 2021a,b; Flynn et al. 2019; Mildenhall et al. 2019, 2020; Wizadwongsa et al. 2021; Zhou et al. 2018] have shown approaches for photorealistic rendering of complex captured real-world scenes. These methods typically operate by solving for spatially varying color and opacity in a bounded volume, either as a smooth function (e.g., a multi-layer perceptron, as in NeRF [Mildenhall et al. 2020]) or discrete voxel parameters [Chen et al. 2022; Fridovich-Keil and Yu et al. 2022; Sun et al. 2022]. Unfortunately, purely neural approaches require significant amounts of compute for rendering, and even with high-end GPUs, often require tens of seconds to render a single image. Recent works improve upon this speed by baking radiance fields into explicit representations, at the expense of increased memory requirements [Garbin et al. 2021; Hedman et al. 2021; Müller et al. 2022; Reiser et al. 2021; Yu et al. 2021]. These approaches still rely on high-end consumer GPU compute and mostly report real-time rendering on medium resolutions (e.g., 800×800). However, the performance requirements of on-device VR rendering are significantly more prohibitive than those explored in these works. Specifically, compelling rendering on headsets requires high resolution images of 4K or more. On top of this, untethered VR devices such as the Oculus Quest 2 have an order of magnitude less compute power (1.2TFlops) than high-end consumer GPUs (13 and 34 TFlops for RTX 2080, 3080 respectively) and limited memory (6GB vs 11GB-24GB). In this work, we design a volumetric scene representation that can render 4K images in real-time and on-device.

In many prior works, the volumetric representation is optimized to reconstruct the input views, which depending on the scene, can consist of hundreds of individually captured photographs. In this work, we show that a high-resolution volumetric representation can be optimized from a single 360° video, which has enough views to capture the underlying 3D appearance while also being practical.

End-to-end systems. Existing techniques for capturing real-world VR scenes generally fall into two categories: (1) rig-based, and (2) handheld. Rig-based capture techniques rely on sophisticated, well-calibrated capture rigs consisting of many camera sensors [Anderson et al. 2016; Broxton et al. 2020; Overbeck et al. 2018; Pozo et al. 2019]. While these systems are able to capture highly detailed virtual worlds including dynamic scenes, the specialized setup makes these approaches inaccessible for everyday captures and ordinary users.

In contrast, handheld or *casual* capture techniques remove the need for complicated capture hardware, and instead only rely on single commodity devices, like smartphones or 360° video cameras.

In order to capture the necessary information for rendering a realistic virtual reality environment, these methods typically prescribe an exhaustive capture process, such as a panoramic sweep [Bertel et al. 2020; Hedman et al. 2017; Hedman and Kopf 2018]. Unfortunately, the requirement of a specific capture protocol renders many of these techniques inapplicable to previously captured content, *i.e.*, videos which may have been captured without the deliberate intent of VR rendering.

3 APPROACH

Given a single 360° video of a static scene, our goal is to recover an explorable 3D scene that can be rendered in virtual reality on a consumer untethered headset.

This goal poses two main challenges. First, VR headsets impose significant constraints on the representation that can be used for rendering. The requirements for spatial resolution, compute, and memory are well beyond the capabilities of existing real-time volumetric rendering methods. Second, many videos cover large distances, and therefore capture scenes with a large spatial extent. Modeling these large scenes introduces another challenge: scaling the representation while continuing to meet the performance specifications.

3.1 Plenoptic Tunnel

To address the aforementioned challenges, we propose the *plenoptic tunnel*, a novel representation consisting of multiple overlapping local 3D representations. Each local representation consists of a view-dependent multi-sphere image (MSI) that can model a local 3D scene efficiently, both in terms of latency and memory. To represent a larger scene region, we can chain together these MSIs, creating a seamlessly explorable volume along the camera path.

In this section, we first describe the parameterization of the local view-dependent MSI, and then discuss how they can be chained together to represent scenes with a large spatial extent. Then, we discuss the process of optimizing for a plenoptic tunnel from a single 360° video, and finally, we describe the process of rendering a scene interactively on-device.

3.1.1 Local 3D Scene Representation. One fundamental challenge of rendering for VR is that it requires rendering *unbounded* omnidirectional scenes. This "inside-out" rendering setup requires *high angular resolution* in the underlying 3D representation, while also being *fast* to render, and *compact* in memory.

View-dependent MSIs. These requirements are a challenge for existing accelerated volumetric radiance fields that mostly focus on inward-facing (i.e., object-centric) scenes. These approaches accelerate rendering by caching the radiance fields into variants of a 3D voxel grid [Garbin et al. 2021; Hedman et al. 2021; Reiser et al. 2021; Yu et al. 2021]. However, all these methods represent the scene volume with a uniformly spaced grid. This results in low angular resolution for inside-out rendering in VR, particularly for content that is close to the virtual camera. We address this by re-parametrizing the 3D volume as view-dependent multi-sphere images (MSIs). MSIs are a generalization of multi-plane images (MPIs) [Penner and Zhang 2017; Wizadwongsa et al. 2021; Zhou et al. 2018] as spheres: a set of L concentric spheres with radii equidistant in inverse depth. MSIs have the benefit of enabling immersive fields-of-view and higher angular resolution while maintaining low memory overhead in the scene parameterization. Additionally, due to their compact nature (i.e., only sampling L pre-determined depth values instead of dense voxel-grid sampling) they are also significantly faster to render. Note that some recent approaches accelerate rendering time by caching the locations of non-zero density values [Müller et al. 2022; Yu et al. 2021]; however, such methods incur additional non-trivial memory storage.

Specifically, every lth sphere in a view-dependent MSI is represented by a $I_l = H \times W \times (4 + K)$ tensor, where the first four channels store the RGB and alpha values, and the last K channels correspond to view-dependent coefficients used to represent non-lambertian effects. Following Wizadwongsa *et al.* [2021], we use the view-dependent coefficients in conjunction with a set of learnable basis functions $H(\vec{v}) \in \mathbb{R}^{K\times 3}$, modeled using a multi-layer perceptron (MLP). This MLP is used to learn the bases, but the function values are cached after training, such that they can be accessed efficiently during rendering. Our method uses a spatial resolution of 4K (3840 x 1920) for each sphere with L =64 layers and K=3. These values are selected with the performance considerations in mind and are ablated in Section 4.

Compact MSI Parametrization. Although MSIs are a compact representation, at higher resolutions (and with many layers), the number of parameters can still be a limiting factor. Even without view-dependence, the MSI representation at 4K resolution results in approximately 5 times as many parameters as a 512^3 voxel grid, one of the higher resolutions used in recent explicit volumetric radiance fields [Fridovich-Keil and Yu et al. 2022]. For our purposes, this parameterization is prohibitive: the number of parameters are too many to store on device memory, which must maintain the scene's RGB, alpha, and view-dependent coefficients paged-in for immediate rendering. Additionally, this increased parameterization also results in a higher potential for degeneracy during optimization.

A standard approach for reducing the number of parameters is to assume a low-rank representation (as is also usually done with the view dependence term), recently explored in Chen *et al.* [2022], which factorize a voxel grid into a linear combination of vector-matrix (VM) outer products. However, this factorization still only reduces a voxel volume into 192 distinct factorized components. As we show in Section 4, this amount of memory usage and computation (for multiplying and adding these vectors and matrices) is still a bottleneck for rendering on mobile devices.

We find that the volume can instead be factorized using the product of three matrix (MMM) tensors, which has a higher representational power, while maintaining a lower computational and memory cost. Recall that a multi-sphere image is stored as 3D voxel grid consisting of L equirectangular images: a total tensor shape of $H\times W\times L$. In this parametrization, the majority of spatial information is stored in the $H \times W$ axis, where different portions of an image may be at different distances from the camera. Such structures are challenging to model with a VM tensor (*e.g.* a decomposition of HW-plane and L-vector only enables a weighted selection of the entire plane at any depth L), requiring a large number of components to compensate. In contrast, decomposing the space into three matrix triplets

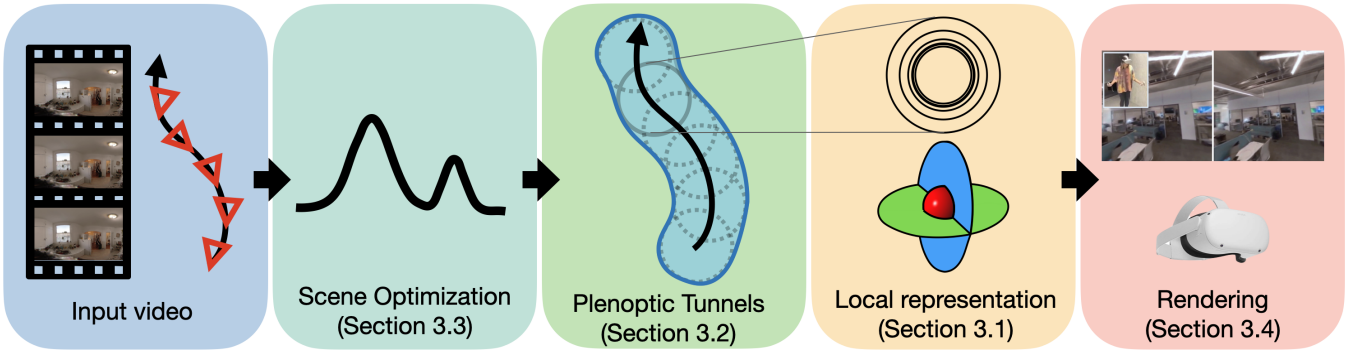


Fig. 2. **Overview of the proposed system.** The input to our system is a single 360° video sequence of a static scene and its estimated camera poses computed via visual SLAM. We then solve for a plenoptic tunnel, a volumetric 3D scene representation that consists of a series of overlapping view-dependent multi-sphere images (MSIs). Each MSI represents the local 3D scene appearance and is parameterized by a low-rank representation deigned to meet the memory and performance needs of on-edge rendering. Given a virtual camera position, we fetch the underlying representation of the nearest MSI and render novel views at interactive rates on consumer VR devices.

can flexibly place different parts of the plane at every axis[1]. In a voxel grid, our factorization basis intuitively forms a 'shell' around the volume. In spherical coordinates, they correspond to the inner most sphere and two cross-sectional planes along the MSIs. This is illustrated in Figure 3.

With MMM tensors, a voxel value at indices x, y, z is computed as

$$f([x,y,z]) = \sum_{k=1}^{K} \left(M_k^{1,2}[x,y] \times M_k^{2,3}[y,z] \times M_k^{1,3}[x,z] \right), \quad (1)$$

where we assume that the indexing operation $\cdot[\cdot]$ incorporates interpolation as needed [Chen et al. 2022]. The volume is then parameterized by $M^{1,2} \in \mathbb{R}^{H \times W}, M^{2,3} \in \mathbb{R}^{W \times L}, M^{1,3} \in \mathbb{R}^{H \times L}$ matrices for K matrix triplet components.

In our model, we have a total of seven volumes: one for alpha, three for base diffuse color, and three for view-dependency. We use MMM factorization for the diffuse and view-dependency volumes. We find that the alpha volume benefits from stronger regularization, and therefore use VM with 6 components, similar to recent explicit volumetric radiance methods [Chen et al. 2022; Fridovich-Keil and Yu et al. 2022]. With the low-rank factorization, there is no need to store an entire volume representing the MSIs, but only the MMM and VM bases. We pack these basis functions as an atlas and compute the voxel features on the fly in the shader.

3.1.2 Modeling Long Trajectories. While the view-dependent MSI is compact and enables 6DoF novel-view synthesis, the regions in which it can synthesize views are limited to the diameter of the innermost sphere. To extend this viewable region, we represent the scene using a number of overlapping MSIs that cover the camera trajectory. At render time, as the user moves through the space, we simply switch to the closest MSI, given on-device constraints. The resulting representation is a 6DoF explorable region along the input camera trajectory, hence we call our representation a *plenoptic tunnel*. In order to ensure consistency between MSIs, and therefore smooth transitions, we optimize for MSIs that are overlapping, and thus share training input images.

[1]Note that the inflexibility of VM tensor can also serve as a form of regularization.

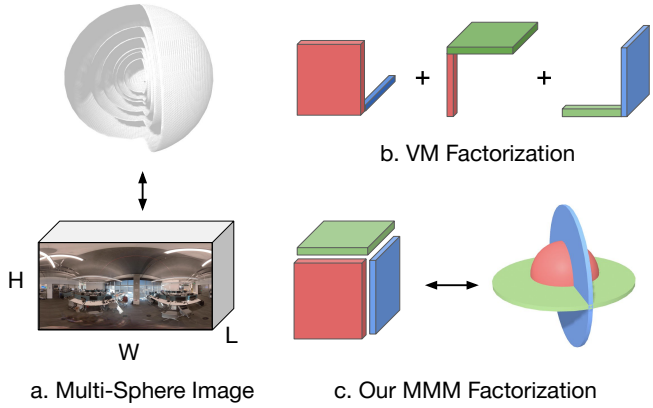


Fig. 3. **MSI Factorization.** (a) MSIs are equivalent to a 3D voxel grid consisting of a set of equirectangular images. In order to compress this 3D volume, we parametrize it using low-dimensional basis functions consisting of three 2D tensors, whose outer product define the volume. Compared to existing VM factorization (b), our MMM factorization (c) is more compact while being more expressive. Intuitively, our basis tensors forms a 'shell' around the volume, or in spherical coordinates correspond to the inner most sphere, and two cross-sectional planes along the layers (right).

3.2 Optimizing Plenoptic Tunnels

We recover a Plenoptic Tunnel representation from a single 360° video, with known camera poses, by optimizing all view-dependent MSIs to reconstruct the input video. Specifically, to compute the color of any ray $r(t) = O + tD$, we intersect the ray with all MSI layers to obtain a collection of distances from the camera origin:

$$\{t_i : \|r(t_i) - p_i\|_2^2 = R_i^2\}_{i=1}^{L}, \quad (2)$$

where the ith spherical layer has center p_i and radius R_i. At every voxel with position $r(t_i)$, our representation stores alpha α, base diffuse color $c \in \mathbb{R}^3$, and basis coefficients $k_i \in \mathbb{R}^3$ [Wizadwongsa et al. 2021]. To render a pixel we first compute the accumulated

(a) Office (28.08 dB) (b) Double Arch (25.04 dB) (c) Roof (28.94 dB) (d) Canyon (24.91 dB)

(e) Living Room (33.81 dB) (f) Hike (23.65 dB) (g) Bedroom (29.86 dB) (h) Kitchen (31.50 dB)

(i) Matterhorn (24.40 dB) (j) Lake Baikal (25.93 dB) (k) Cinque Terre (23.67 dB) (l) Castle (27.57 dB)

Fig. 4. **Datasets.** A visualization of the datasets used for evaluating our method. The bottom four sequences (i-l) are previously captured videos downloaded from the Internet (credit:AirPano VR). We report the held-out PSNR performance for each video.

alpha weights across the first i layers, $w(i)$:

$$w(i) = \alpha(r(t_i))\exp\left(-\sum_{j=1}^{i}\alpha(r(t_j))\right). \tag{3}$$

Note that in contrast to continuous volumetric representations such as NeRF [Mildenhall et al. 2020], our render is discretized rather than continuous, eliminating the need to sample exhaustively to approximate continuous values.

Then, the view-dependent color along the ray is accumulated as a weighted combination using $w(i)$ to obtain the final ray color $\hat{C}(r)$:

$$\hat{C}(r) = \sum_{i=1}^{L} \underbrace{w(i)}_{\text{weight}} \underbrace{[c(r(t_i))}_{\text{diffuse}} + \underbrace{k(r(t_i))^T H(D)]^+}_{\text{specular}}. \tag{4}$$

where $c(\cdot)$ and $k(\cdot)$ are the sampled diffuse and view-dependent color, and $H(\cdot) \in \mathbb{R}^{K\times 3}$ is the basis functions of size K.

Objective. With this differentiable rendering function, we optimize all MSI parameters to minimize the total squared error between the rendered and true pixel colors:

$$\mathcal{L} = \sum_{r\in\mathcal{R}} ||\hat{C}(r) - C(r)||_2^2. \tag{5}$$

We additionally utilize a total variation regularization within each sphere similarly to [Wizadwongsa et al. 2021].

On-Device Demo. The on-device demo is built in three.js, which is in turn built on top of WebGL and WebXR. This makes our deployed 6DoF explorable experiences device and platform-agnostic. To export the trained Plenoptic Tunnel, we pack all textures into a 16384x16384 texture atlas. This texture atlas is loaded directly into a minimal precision `Uint8Array` array before being transferred to GPU in a *low precision* 24-bit format. Our custom shader then computes the voxel features it needs on-demand. Please see the supplemental for more implementation details.

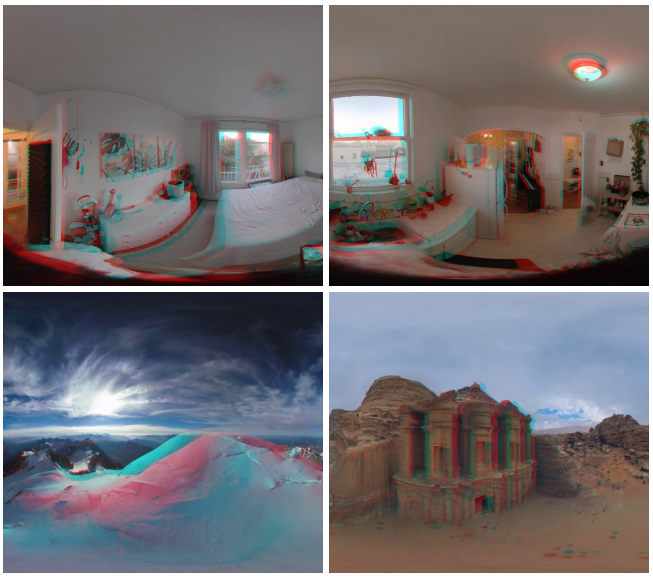


Fig. 5. **Anaglyphs.** A visualization of our rendered representations as anaglyphs, demonstrating that our reconstructions estimate plausible scene structure and simulate realistic parallax.

4 EXPERIMENTS

We evaluate our method on a diverse collection of twelve 360° video sequences, eight self-captured and four downloaded from the Internet. A frame from each input sequence is shown in Figure 4.

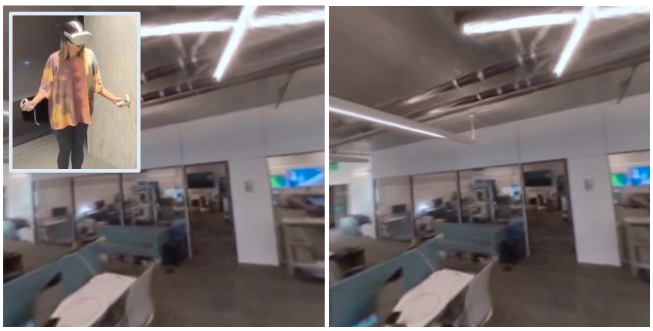


Fig. 6. **Interactive demo.** A visualization of our interactive virtual reality demo, rendering at 90Hz and 4K on an Oculus Quest 2.

We also evaluate our approach on the OmniPhotos [Bertel et al. 2020] sequences. We furthermore ablate key design decisions that significantly impact memory and compute on-device: hyperparameters including layer count and factorization strategy. Please see the supplemental video for more qualitative results.

Qualitative results. In Figure 7, we show a collection of novel views synthesized by our method along with the corresponding ground truth held-out views. We also synthesize a viewpoint off the captured trajectory to show that our method generalizes beyond the input trajectory. Our approach is able to handle a wide variety of indoor and outdoor scenes, ranging from cluttered office scenes to distant landscapes, and can even realistically reproduce hard-to-capture materials including reflections and transparency (see video). Note that the images shown in the paper are produced through the same rendering pipeline used in our interactive VR demo. In the supplemental video, we provide a video of a user exploring a scene in our interactive demo. In Figure 5 we render several scenes using anaglyphs illustrating the parallax captured and simulated by our system.

	L	VR?	FLOPs/pt	FPS	Mem.	PSNR↑	LPIPS↓
Sph-NeX	192	✗	26	–	18.4 GB	27.51	0.3678
Sph-NeX	64	✗	26	–	14.8 GB	25.32	0.3968
Ours-Big	192	✗	72	43.9	**32.80 MB**	27.80	0.3653
Ours-VM3	64	✗	375	4.7	**28.79 MB**	27.52	0.3684
Ours-SH	64	✓	72	**90**	**28.75 MB**	27.41	0.3679
Ours	64	✓	72	**90**	**28.75 MB**	28.22	0.3547

Table 1. **Ablation study.** We ablate our design choices to find that (1) the final model does not compromise on quality compared to big models that do not fit or run on-device in real-time (and in fact, it performs best), (2) The proposed MMM factorization reduces FLOPs considerably while achieving slightly higher PSNR. Interactive on-device rendering requires 90+ FPS, and memory under 6GB. Models that satisfy both these constraints are shown as capable of rendering VR content. Please see the text for details.

Quantitative Evaluations. To validate the design decisions in our system, we conduct an ablation study on the nine self-captured videos, reported in Table 4. We hold out every 8th frame during training and use that frame to evaluate the novel-view synthesis quality using PSNR and LPIPS [Zhang et al. 2018]. To measure computational cost and system performance, we report the number of floating point operations (FLOPs) needed to render color for a single point in space and the frames per second (*FPS*) measured on an Oculus Quest 2. To mitigate the effects of motion sickness, we need to achieve at least 90 FPS indicated in bold. In addition, we include the memory consumption (*Mem*) for the texture atlas, in `float32`, as quantized weights decrease PSNR moderately (1-2 dB) on-device. For reference, the total available RAM on Oculus Quest 2 total is 6GBs; we bold methods that consume less memory. In order to render at interactive framerates on device, both the $\texttt{FPS} \geq 90$ and $\texttt{Mem} \leq 6GB$ constraints must be met.

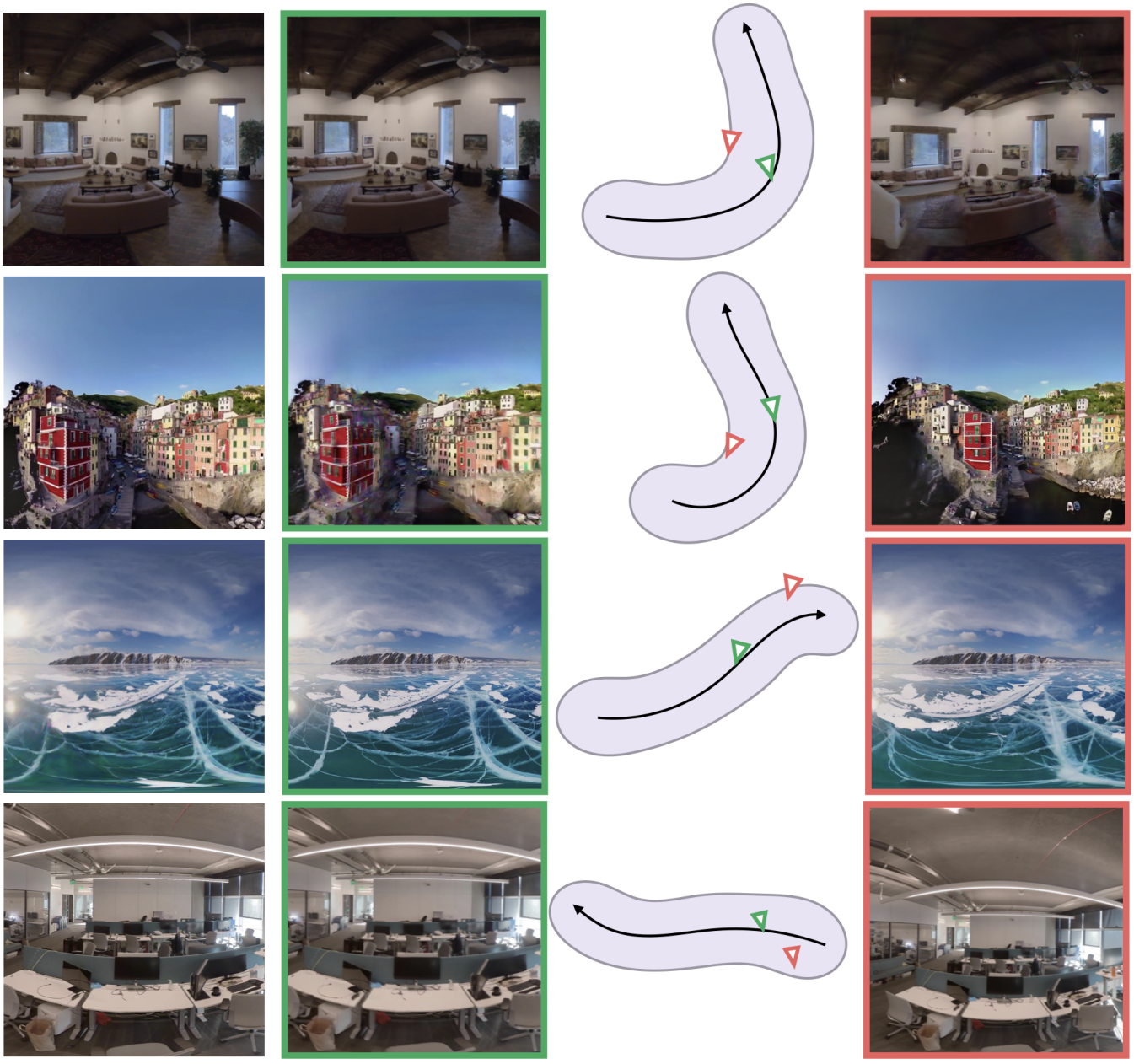
We first compare our approach with the closest prior work: a variant of NeX [Wizadwongsa et al. 2021], a method for optimizing view-dependent multi-plane images. To adapt NeX to our problem setting, we change the formulation from multi-plane images to multi-sphere images. We use the same hyperparameters, namely 192 layers and 8 learned basis functions. While both the rendering time and memory consumption of this approach is well beyond what could be reasonably used for real-time VR rendering, this model, Sph-NeX, serves as a baseline for a "high-quality" but non-interactive method. One may attempt to reduce the complexity of the underlying geometry by reducing the layers to 64, with hopes that the simpler representation may have a small enough memory footprint to fit on device. This model, however, (Sph-Nex-L64) results in a significant drop in PSNR quality. Furthermore, when compared to our model with the same number of layers, we find that our design choices achieve better visual quality while better satisfying the performance requirements for on-device rendering. This is likely because our factorization scheme drastically reduces the number of parameters, resulting in a better underlying 3D representation. We also evaluate our approach using degree 1 spherical harmonics, Ours-SH, which incurs small reduction in quality.

We additionally compare our proposed factorization technique to alternatives proposed in prior work [Chen et al. 2022] (labeled as Ours-VM3), where we use VM instead of MMM factors keeping all else equal. We find that MMM lead to slightly better PSNR, while reducing the number of FLOPs by a factor of 10. MMM also has $O(n)$ fewer parameters than VM. Note our method (last row) attains the highest performance despite also being the fastest (i.e., being able to render on-device).

Lastly, we evaluate our method on eight of the OmniPhotos [Bertel et al. 2020] sequences. Although the original paper does not provide quantitative evaluation on these sequences, we develop a quantitative validation scheme for evaluating the quality of our rendered novel views against held-out images. We will commit to releasing this evaluation protocol to facilitate future research in this domain. On this dataset our approach obtains average PSNR of 25.04. Please see the supplemental for per-scene metrics.

5 CONCLUSION AND FUTURE WORK

We present an end-to-end solution for converting a single 360° video into an immersive 6DoF experience viewable in real-time on an untethered VR headset. While this method enables immersive exploration of captured videos, it assumes that the captured scene is static. A promising direction for future work would be to extend this technique to allow exploration of video content.



(a) Ground truth held-out view (b) Rendered held-out view (c) Off-trajectory viewpoint (d) Capture trajectory

Fig. 7. **Results.** A collection of results showing the quality of our synthesized novel views. Along the input capture path, our rendered images (second row) are able to accurately reproduce appearance of the corresponding real frames (first row). Additionally, our method can render unseen viewpoints further away from the captured trajectory (third row) without additional artifacts. In the last row, we show a top-down visualization of the camera path with highlighted viewpoint frusta.

Additionally, our method inherits many of the characteristics of volumetric reconstruction techniques, such as the need for multiple viewpoints. In cases where the input video contains purely linear camera motion, e.g. moving the camera in a straight line, scene content that is colinear with the camera trajectory may not have accurately reconstructed geometry, and can therefore have less realistic appearances when rendering viewpoints further from the capture trajectory.

REFERENCES

Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski. 2011. Building rome in a day. *Commun. ACM* 54, 10 (2011), 105–112.

Robert Anderson, David Gallup, Jonathan T Barron, Janne Kontkanen, Noah Snavely, Carlos Hernández, Sameer Agarwal, and Steven M Seitz. 2016. Jump: virtual reality video. *ACM Transactions on Graphics (TOG)* 35, 6 (2016), 1–13.

Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. 2021a. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 5855–5864.

Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. 2021b. Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields. *arXiv preprint arXiv:2111.12077* (2021).

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

Tobias Bertel, Mingze Yuan, Reuben Lindroos, and Christian Richardt. 2020. OmniPhotos: casual 360° VR photography. *ACM Transactions on Graphics (TOG)* 39, 6 (2020), 1–12.

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

Chris Buehler, Michael Bosse, Leonard McMillan, Steven Gortler, and Michael Cohen. 2001. Unstructured lumigraph rendering. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. 425–432.

Anpei Chen, Zexiang Xu, Andreas Geiger, , Jingyi Yu, and Hao Su. 2022. TensoRF: Tensorial Radiance Fields. arXiv:2203.09517 [cs.CV]

John Flynn, Michael Broxton, Paul Debevec, Matthew DuVall, Graham Fyffe, Ryan Overbeck, Noah Snavely, and Richard Tucker. 2019. Deepview: View synthesis with learned gradient descent. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2367–2376.

Fridovich-Keil and Yu, Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. 2022. Plenoxels: Radiance Fields without Neural Networks. In *CVPR*.

Yasutaka Furukawa and Carlos Hernández. 2015. Multi-view stereo: A tutorial. *Foundations and Trends® in Computer Graphics and Vision* 9, 1-2 (2015), 1–148.

Stephan J Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. 2021. Fastnerf: High-fidelity neural rendering at 200fps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 14346–14355.

Peter Hedman, Suhib Alsisan, Richard Szeliski, and Johannes Kopf. 2017. Casual 3D photography. *ACM Transactions on Graphics (TOG)* 36, 6 (2017), 1–15.

Peter Hedman and Johannes Kopf. 2018. Instant 3d photography. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–12.

Peter Hedman, Tobias Ritschel, George Drettakis, and Gabriel Brostow. 2016. Scalable Inside-Out Image-Based Rendering. 35, 6 (2016), 231:1–231:11.

Peter Hedman, Pratul P. Srinivasan, Ben Mildenhall, Jonathan T. Barron, and Paul Debevec. 2021. Baking Neural Radiance Fields for Real-Time View Synthesis. *ICCV* (2021).

Jingwei Huang, Zhili Chen, Duygu Ceylan, and Hailin Jin. 2017. 6-DOF VR videos with a single 360-camera. In *2017 IEEE Virtual Reality (VR)*. IEEE, 37–44.

Kiriakos N Kutulakos and Steven M Seitz. 2000. A theory of shape by space carving. *International journal of computer vision* 38, 3 (2000), 199–218.

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

Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. 2019. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–14.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *ECCV*.

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. 2022. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *ACM Trans. Graph.* 41, 4, Article 102 (July 2022), 15 pages. https://doi.org/10.1145/3528223.3530127

Ryan S Overbeck, Daniel Erickson, Daniel Evangelakos, Matt Pharr, and Paul Debevec. 2018. A system for acquiring, processing, and rendering panoramic light field stills for virtual reality. *ACM Transactions on Graphics (TOG)* 37, 6 (2018), 1–15.

Jeong Joon Park, Aleksander Holynski, and Steven M Seitz. 2020. Seeing the world in a bag of chips. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1417–1427.

Eric Penner and Li Zhang. 2017. Soft 3D reconstruction for view synthesis. *ACM Transactions on Graphics (TOG)* 36, 6 (2017), 1–11.

Marc Pollefeys, Luc Van Gool, Maarten Vergauwen, Frank Verbiest, Kurt Cornelis, Jan Tops, and Reinhard Koch. 2004. Visual modeling with a hand-held camera. *International Journal of Computer Vision* 59, 3 (2004), 207–232.

Albert Parra Pozo, Michael Toksvig, Terry Filiba Schrager, Joyce Hsu, Uday Mathur, Alexander Sorkine-Hornung, Rick Szeliski, and Brian Cabral. 2019. An integrated 6DoF video camera and system design. *ACM Transactions on Graphics (TOG)* 38, 6 (2019), 1–16.

Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. 2021. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 14335–14345.

Johannes Lutz Schönberger and Jan-Michael Frahm. 2016. Structure-from-Motion Revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. 2016. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision*. Springer, 501–518.

Shinya Sumikura, Mikiya Shibuya, and Ken Sakurada. 2019. Openvslam: a versatile visual slam framework. In *Proceedings of the 27th ACM International Conference on Multimedia*. 2292–2295.

Cheng Sun, Min Sun, and Hwann-Tzong Chen. 2022. Direct Voxel Grid Optimization: Super-fast Convergence for Radiance Fields Reconstruction. *CVPR* (2022).

Shimon Ullman. 1979. The interpretation of structure from motion. *Proceedings of the Royal Society of London. Series B. Biological Sciences* 203, 1153 (1979), 405–426.

Suttisak Wizadwongsa, Pakkapon Phongthawee, Jiraphon Yenphraphai, and Supasorn Suwajanakorn. 2021. NeX: Real-time View Synthesis with Neural Basis Expansion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Daniel N Wood, Daniel I Azuma, Ken Aldinger, Brian Curless, Tom Duchamp, David H Salesin, and Werner Stuetzle. 2000. Surface light fields for 3D photography. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. 287–296.

Changchang Wu et al. 2011. VisualSFM: A visual structure from motion system. (2011).

Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. 2021. PlenOctrees for Real-time Rendering of Neural Radiance Fields. In *ICCV*.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *CVPR*.

Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. 2018. Stereo Magnification: Learning View Synthesis using Multiplane Images. In *SIGGRAPH*.