TeTriRF: Temporal Tri-Plane Radiance Fields for Efficient Free-Viewpoint Video

Minye Wu Zehao Wang Georgios Kouros Tinne Tuytelaars
KU Leuven

Figure 1. **Left**: Temporal Tri-Plane Radiance Fields structure with plane-grid hybrid representation. **Middle**: TeTriRF results, showcasing novel view synthesis in dynamic scenes. **Right**: Comparison with existing methods in storage, speed, and quality, where larger circles indicate better quality.

Abstract

Neural Radiance Fields (NeRF) revolutionize the realm of visual media by providing photorealistic Free-Viewpoint Video (FVV) experiences, offering viewers unparalleled immersion and interactivity. However, the technology's significant storage requirements and the computational complexity involved in generation and rendering currently limit its broader application. To close this gap, this paper presents Temporal Tri-Plane Radiance Fields (TeTriRF), a novel technology that significantly reduces the storage size for Free-Viewpoint Video (FVV) while maintaining low-cost generation and rendering. TeTriRF introduces a hybrid representation with tri-planes and voxel grids to support scaling up to long-duration sequences and scenes with complex motions or rapid changes. We propose a group training scheme tailored to achieving high training efficiency and yielding temporally consistent, low-entropy scene representations on feature domain. Leveraging these properties of the representations, we introduce a compression pipeline with off-the-shelf video codecs, achieving an order of magnitude less storage size compared to the state-of-the-art. Our experiments demonstrate that TeTriRF can achieve competitive quality with a higher compression rate.

1. Introduction

Advanced VR/AR devices are boosting interest in Free-Viewpoint Video (FVV), which allows users to choose their own viewing angles for a unique and immersive exploration experience. The emergence of Neural Radiance Fields (NeRF), as introduced in [24], has significantly advanced FVV demonstrating unprecedented photorealism in rendering. However, one main challenge with this technology, apart from its rendering speed, is the extensive storage space required for preserving reconstructed 4D data. This requirement complicates the process of transferring and storing such data on user devices, making the creation and use of long sequence FVV increasingly impractical.

Recent advances in NeRF facilitate dynamic scene rendering for FVV generation. Some models [8, 22, 26, 29] use deformation fields to model scene motion, mapping each frame to a canonical space. While these capture dynamics effectively, they are constrained by the high computational load of implicit feature decoding [8, 26, 29] or by the large storage needs of explicit 3D grid-based representations [22]. Alternatively, novel radiance field representations have been proposed to record dynamic scenes. They incorporate 4D data using techniques like planar factorization [9], Fourier coefficients [39], and latent embeddings [19]. By training jointly across multiple frames, these methods achieve more efficient sequential frame reconstruction. However, their overly compact representation with limited capacity compromise their performance in capturing complex motions and long sequences. Most recently, several methods [28, 34, 40] have been developed to significantly improve the storage-performance trade-off. However, NeRFPlayer [34] suffers from a notably slow rendering speed, which prevents real-time playback for FVV. Dynamic MLP Maps [28] and ReRF [40], on the other hand,

achieve real-time rendering, but they require the use of a high-end GPU for decoding.

In this paper, we present a novel FVV modeling approach called Temporal Tri-Plane Radiance Fields or TeTriRF, which achieves efficient FVV generation and rendering with extremely compact storage. This is achieved via three main innovations. First, we propose a learning scheme that results in temporally consistent and low-entropy sequential feature representations that can be effectively compressed. At the core is a training strategy that groups consecutive frames from sequential data and reduces the entropy of the frame representations via imposing feature temporal consistency by deploying intra-group and inter-group regularizers. By sharing temporal information during training, TeTriRF is able to dramatically accelerate training compared to the per-frame training methods. We also deploy a two-pass progressive scaling scheme to reduce the cost of preprocessing while enhancing rendering quality and compression rate by discarding noise in empty space.

Second, we introduce a hybrid representation that combines tri-planes with voxel grids for frames within the sequence. Specifically, for each frame in the stream, we factorize the radiance field to a tri-plane and a 3D density grid. This hybrid approach effectively captures high-dimensional appearance features in compact planes and enables efficient point sampling through the explicit density grid, achieving a balance between compactness and representation effectiveness. Building upon this hybrid representation, we adopt a deferred shading model [13, 30] paired with lightweight MLP decoders to bring real-time rendering within reach.

Third, we show how to compress our FVV hybrid representation compactly using off-the-shelf video codecs. To achieve this, we develop a compression pipeline specifically for TeTriRF, which includes processes such as value quantization, removal of empty spaces, conversion into 2D serialization, and subsequent video encoding. The temporally consistent and low-entropy properties of our representation significantly enhance data compression efficiency. With our model, we're able to produce high-quality results with just 10-100 KB/frame. This means that a one-hour video could be stored in 1-10 GB, which is for the first time, within range of memory available on AR/VR devices. TeTriRF, with its compact size and our hybrid representation, is capable of handling long sequence FVV effectively. Together with our lightweight renderer and hardware accelerated video decoding, our approach takes another step towards streaming and rendering photorealistic FVV for end-users.

Fig. 1 illustrates TeTriRF's representation structure and comparison with other methods. TeTriRF distinguishes itself with superior advantages in storage efficiency, rendering speed, and rendering quality.

2. Related Work

Neural Scene Representations. NeRF [24] achieves photo-realistic novel view synthesis using a simple implicit representation. Despite the quality and compactness of NeRF, the scene reconstruction and rendering times are substantial and prohibitive for the reconstruction of both static and dynamic scenes. Subsequent works distill the volumetric representation of the scene into voxel grids to achieve real-time rendering speeds [13, 23, 45] or also fast volumetric scene reconstruction [32, 35]. Nonetheless, these approaches encounter the drawback of increased storage size due to the use of 3D grid representations. These issues have partly been mitigated by substituting the 3D voxel grid representation with more compact and memory-efficient tensor decompositions [4]. Further improvements in training and rendering speed have also been achieved by leveraging trainable multi-resolution hash tables [25] or representations based on 3D Gaussians [16]. While efficient and compact photo-realistic reconstruction and rendering for static scenes are now achievable, the main challenge remains in dynamic scenes. In such scenarios, storage requirements typically increase linearly with the number of frames, leading to greater difficulties.

Free-Viewpoint Video Representations. Previous works use textured animated meshes [5, 20, 21] or layered meshes [1], which consist of temporally coherent geometries for compression, to represent FVV content, but the mesh-based representations limit their rendering quality. The other methods [36, 47] utilize a neural blending approach to achieve photo-realistic rendering based on meshes. NeRF-based methods have significantly enhanced rendering quality. Some approaches reconstruct dynamic scenes by conditioning an implicit representation on time [7, 10, 43] or time-varying latent codes [18, 28]. Alternatively, other methods [7, 8, 18, 26, 27, 29, 37, 46] optimize a deformation field to predict the displacement of the scene across time between each frame and a reference canonical frame. Most implicit time-conditioned or deformation-based methods suffer from slow training and rendering speeds. To accelerate the speeds, methods have been developed using grid representations [12, 22], 4D plane-based representation [3, 33, 44], and tensor factorization [2, 9, 15]. Even though they provide faster training or rendering times but they usually suffer from storage efficiency or network capacity issues. Our proposed method resembles K-planes [9], a method based on planar factorization. However, rather than using an additional tri-plane for spatio-temporal variations we simply unfold a triple-plane representation along the temporal dimension. Combined with our training and compression schemes, our method achieves improved rendering speed and compactness with competitive quality.

Neural Radiance Field Compression. Implicit neural

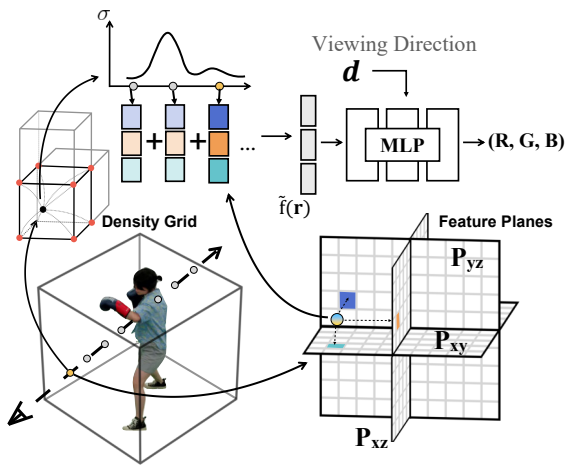

Figure 2. **Illustration of the Hybrid Representation.** Our Hybrid Tri-Plane approach models each frame using a density grid and a tri-plane. We adopt the deferred shading model in our rendering pipeline.

representations like NeRF [24] are relatively compact but at the same time extremely slow to train and render. Improved compression in the context of neural radiance fields has been achieved via vector quantization [17], wavelet transforms on grid-based neural fields [31], parameter pruning [6], and Fourier transform [14]. These approaches are primarily focused on static scenes and lack the capability to compress temporal information. Recent works extend the compression into dynamic scene by using tensor decomposition [34] , residual radiance fields with specialized video codecs [41], and reducing spatio-temporal redundancies of feature grids [11]. However, fast decoding and rendering time is still an issue with these methods. Our method, on the other hand, achieves both high compression and fast decoding and rendering times thanks to our proposed representation, lightweight renderer, and off-the-shelf encoding.

3. Methodology

Our method is able to generate free-viewpoint videos from synchronized multi-view video inputs to support novel view synthesis in dynamic scenes with long-duration sequences and complex motions. We represent each frame using a specialized hybrid representation with disentangled and compactly organized geometry and appearance, complemented by an efficient rendering pipeline. We propose a fast training scheme (Sec. 3.2) that trains multiple frames in groups and lowers the entropy of features by improving their temporal consistency. This facilitates extremely compact compression in our proposed pipeline. We demonstrate that these representations can be efficiently compressed using off-the-shelf video codecs (Sec 3.3).

3.1. Hybrid Tri-Plane

In this work, as illustrated in Fig. 2, we use a hybrid representation composed of a 3D density grid $\mathbf{V}_\sigma$ and a feature Tri-Plane $\mathbf{P} = \{\mathbf{P}_s | s \in \mathcal{S}\}$ to represent each frame, where each element in $\mathbf{P}$ is a 2D grid with $h = 10$ channels and $\mathcal{S} = \{xy, xz, yz\}$. The purpose of this hybrid design is to attain a good trade-off between effectiveness and compactness. The utilization of an explicit density grid allows the direct and fast acquisition of density values. This enables the straightforward construction of a grid mask to efficiently discard sample points in free space without network inference and hence speed up both training and rendering. A feature Tri-Plane, on the other hand, contains three orthogonal feature planes factorizing the spatial space of higher-dimensional appearance features. We adopt the compactness of this plane-based representation, which has been shown in [9], to elevate the compression rate to a new level.

Rendering. TeTriRF queries the density σ of a 3D point $\mathbf{x}$ by applying trilinear interpolation $\varphi_t(\cdot)$ on the 3D density grid V_σ. Appearance features, on the other hand, are acquired by projecting the point onto the three feature planes $\mathbf{P}$ and applying bilinear interpolation $\varphi_b(\cdot)$ for each 2D projection. These operations are formulated as:

$$\begin{aligned} \sigma &= \varphi_t(\mathbf{x}, V_\sigma) \\ \mathbf{f}_s &= \varphi_b(\mathbf{x}, \pi_s(\mathbf{P}_s)), \end{aligned} \tag{1}$$

where π_s is a function that projects a 3D point onto plane s, and $\mathbf{f}_s$ is the fetched h-dimensional feature vector from that plane. Then we concatenate the appearance features from the three planes to compose the feature vector $\mathbf{f} = [\mathbf{f}_s | s \in \mathcal{S}]$.

Since having the explicit density grid, we adopt the masking mechanism from DVGO to discard points in empty space and thus reduce the associated overhead in acquisition, processing, and volume rendering. Inspired by [13, 30], we also adopt the deferred shading model which performs volume rendering on the appearance features of sample points along a ray, rather than their radiance. The radiance value (pixel RGB color) $\mathbf{c}$ of the ray is then decoded through a shallow MLP:

$$\mathbf{c}(\mathbf{r}) = \Phi(\tilde{\mathbf{f}}(\mathbf{r}), \omega(\mathbf{d})), \tag{2}$$

where Φ denotes the decoding operation performed by the shallow MLP, responsible for transforming appearance features into radiance and $\tilde{\mathbf{f}}(\mathbf{r})$ is the integrated feature vector form the deferred shading model. Additionally, $\mathbf{d}$ denotes the viewing direction of the ray and $\omega(\cdot)$ represents the positional encoding used in [24]. This deferred rendering approach is also differentiable as gradients are back-propageted from the RGB color to the tri-plane, density grid, and MLP network parameters.

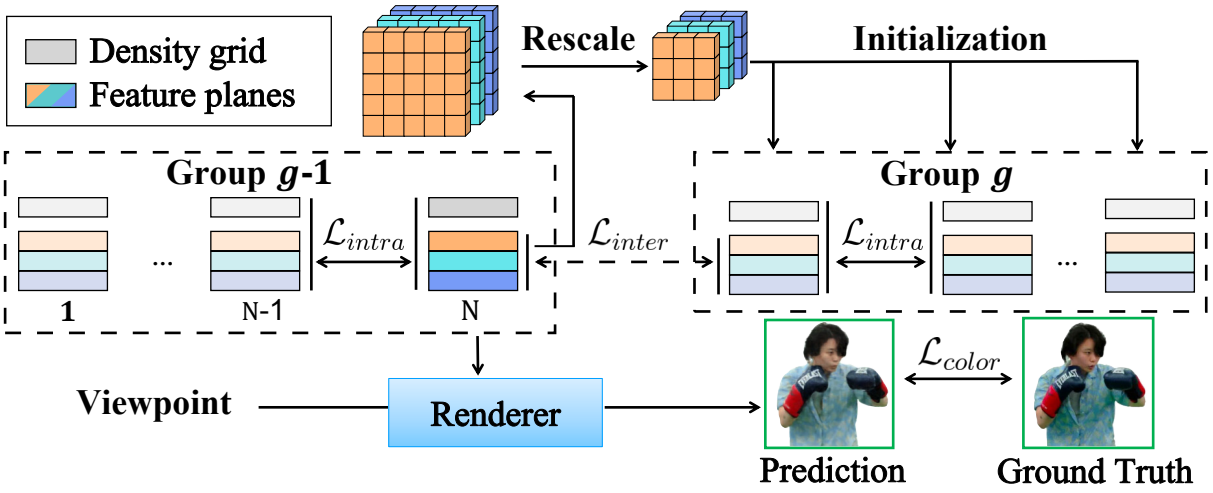

Figure 3. **Grouped Multi-frame Training Overview.** TeTriRF group frames in sequential data and trains a group of N frames together. We deploy photometric loss $\mathcal{L}_{color}$, intra-group loss $\mathcal{L}_{intra}$, and inter-group loss $\mathcal{L}_{inter}$ among the frames. For a given viewpoint, the Renderer (Sec. 3.1) takes the corresponding frame's representation to synthesize novel views.

3.2. Grouped Multi-frame Training

Acquiring hybrid Tri-Plane representations for sequential frames is a non-trivial task. Per-frame training, as employed in ReRF [40], lacks efficiency. This is mainly because ReRF only exploits information from adjacent frames In our method, we train a group of N consecutive frames jointly at one time to more effectively leverage temporal information.We train sequential data in non-overlapping groups according to the timeline, enabling TeTriRF to support long or even infinite sequences. To promote information sharing and reduce redundancy during optimization we regularize the groups at both the intra- and inter-level as detailed in the following paragraphs.

Intra-Group Regularization. We apply L1 loss on density grids and feature planes between adjacent frames to encourage sparsity and minimize density and feature changes. This is crucial because video codecs are tasked with encoding the differences between frames. Consequently, sparsifying and minimizing these changes, effectively reduces the bitrate required for video encoding, leading to more efficient data compression. Temporal information, such as cross-frame density variations, can also be passed and shared among frames resulting in faster training. We formulate this intra-group regularization as:

$$\mathcal{L}_{intra} = \sum_{i=1}^{N-1} \left[\left\| V_\sigma^{g,i} - V_\sigma^{g,i+1} \right\|_1 + \sum_{s \in \mathcal{S}} \left\| P_s^{g,i} - P_s^{g,i+1} \right\|_1 \right], \tag{3}$$

where g and i denote the group index of these frames and the frame index inside the group, respectively. Furthermore, sharing the MLP decoder within the group defines a shared appearance feature space that facilitates faster convergence.

Inter-Group Regularization. To reduce calculation redundancy and speed up optimization, we initialize every frame in the current group with the feature planes from the last frame of the previous group as shown in Fig. 3. This approach leverages existing information to provide an advantageous starting point for the training process. The MLP decoder is also initialized with the parameters of the decoder from the previous group for the same reason.

In addition to the initialization, we also apply an L1 feature loss between the first frame of the current group and the last frame of the previous group, ensuring feature continuity between the groups and thus increasing the compression rate. This L1 cross-group feature loss is formulated as:

$$\mathcal{L}_{inter} = \left\| V_\sigma^{g-1,N} - V_\sigma^{g,1} \right\|_1 + \sum_{s \in \mathcal{S}} \left\| P_s^{g-1,N} - P_s^{g,1} \right\|_1 . \tag{4}$$

We block the gradients to $V_\sigma^{g-1,N}$ and $P_s^{g-1,N}$ from the previous group to keep their representation consistent after being trained. This allows to train one group of frames at a time.

Along with the photometric loss $\mathcal{L}_{color}$, the total loss $\mathcal{L}_{total}$ is calculated as:

$$\mathcal{L}_{total} = \mathcal{L}_{color} + \lambda_1 \mathcal{L}_{intra} + \lambda_2 \mathcal{L}_{inter}, \tag{5}$$

where λ_1 and λ_2 are the weights of the regularizing terms.

Two-pass Progressive Scaling. DVGO [35] and ReRF [41] follow a two-stage coarse-to-fine training scheme. In the first stage, diffuse color is used to reconstruct a coarse density field for building a grid mask used to discard empty space. This strategy helps prevent the occurrence of numerous floaters and enhances the training speed. Likewise, HumanRF [15] precomputes an occupancy grid for making training more efficient. However, the occupancy calculation

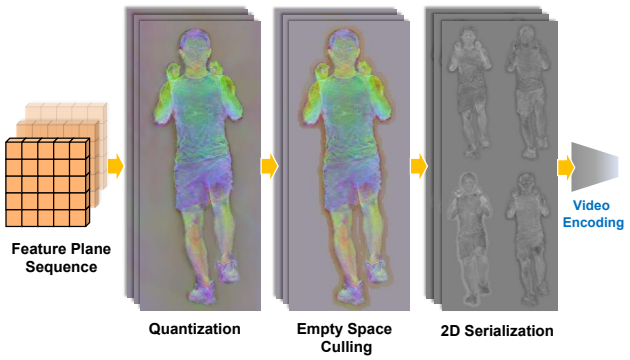

Figure 4. Demonstration of the proposed compression pipeline and the visualizations after each process on a feature plane sequence. Colorized by the first three principal components.

highly depends on the camera setups, requiring the number of visible views for each space to be evenly distributed. All these extra procedures take extra time and resources to process.

In our approach, we also maintain occupancy grids as in [15, 41], but we make them more adaptive and efficient for dynamic scenes regardless of the camera setups. To this end, we introduce a two-pass progressive scaling strategy, where at predefined iterations we rescale the resolution of the density grid and feature planes as in [35]. The first pass upscales the space at shorter time intervals, functioning similarly to the coarse training stage in scene space exploration. Upon completing the first pass, we revert the scale to its initial, lowest resolution. This reduction in resolution diminishes the presence of floaters, thereby increasing the availability of empty space. Most importantly, as we maintain the same MLP decoder throughout the training process, the down-scaled feature planes from the first pass can be effectively reused. This serves as an effective initialization for subsequent training phases, diverging from previous coarse-to-fine training strategies where appearance features are typically retrained from scratch. Following the second progressive scaling pass, we update the training rays by filtering out those that do not hit occupied space inside the bounding box, thus focusing the training on the reconstruction of fine detail. These training strategies can efficiently remove empty space regardless of camera setups and improve results with the same number of iterations as the fine stage of ReRF even without a coarse training stage.

3.3. Efficient FVV Generation

In our methodology, we aim to produce compact FVV (Free-Viewpoint Video) content. This is achieved by encoding the representations, which have been previously trained, in a highly efficient manner. To facilitate this, we employ established, commercial video codecs known for their efficiency. The process involves a necessary transformation of our representations, ensuring their compatibility with the adopted video codecs.

We start by linearly normalizing the numbers so they fall between 0 and 1. According to our statistical analysis, in most scenarios, 99.5% of density and feature values fall in the ranges $[-5, 30]$ and $[-20, 20]$, respectively. Any numbers that fall outside of these ranges are clipped to 0 or 1. Following this, we quantize the normalized values into 12-bit integers. We use the density activation function from DVGO paired with a threshold τ_α to generate a grid mask and then cull empty space from density grids by setting the value in the mask to zero. This eliminates unnecessary temporal changes. Since, feature planes have multiple channels, while compressed video has only a single channel, we flatten (or re-arrange) each channel of a feature plane onto a 2D single-channel image while preserving the 2D spatial continuity. Fig. 4 illustrates our compression pipeline.

In summary, we will have four images for each frame, representing density, xy-plane, xz-plane, and yz-plane, respectively, where each type forms an image sequence that we compress using a video codec. Rendering an FVV also requires the weights of the decoding MLPs for all frame groups. Therefore, we quantize those weights into 16 bits and store them directly without additional compression, given that the size of the MLPs is already relatively small. One benefit of leveraging off-the-shelf video codecs is that there are a lot of available options for software and hardware acceleration that can facilitate the efficient decoding of this kind of content. In TeTriRF, we use the High Efficiency Video Coding (HEVC), also known as H.265, to compress feature images. The H.265 video encoder calculates feature residuals between frames and transforms them into the frequency domain. It employs various quantization parameters to measure coefficients of different frequencies, resulting in quantization that can save space but may lead to some information loss. We adjust Constant Rate Factor (CRF), which is a quality-control setting in video encoding that balances video quality and file size.

4. Experiments

We begin by comparing our method both quantitatively and qualitatively with previous works (Sec. 4.1). Subsequently, we present extensive ablation studies to validate the components we propose (Sec. 4.2). The default group size is $N = 20$. For more detailed information and additional results, please refer to our supplementary materials.

In our experiments, we use three datasets: NHR [42], ReRF [41], and DyNeRF [18]. The first two consist of human-centric dynamic scenes, while DyNeRF contains forward-facing dynamic scenes.

4.1. Comparison

For the human-centric scenes, our method is contrasted with four contemporary dynamic NeRF techniques: KPlanes

	NHR Dataset						ReRF Dataset					
	PSNR ↑	SSIM ↑	LPIPS ↓	T.T. ↓	R.T. ↓	Size ↓	PSNR ↑	SSIM ↑	LPIPS ↓	T.T. ↓	R.T. ↓	Size ↓
KPlanes	30.18	0.963	0.063	0.65	2.2	2986	27.81	0.946	0.094	0.65	2.2	2986
HumanRF	31.91	0.872	0.036	1.4	3.95	2852	28.58	0.876	0.072	1.4	3.95	2852
TiNeuVox	30.45	0.962	0.077	2.4	18.14	5580	28.86	0.947	0.082	2.8	23.65	5580
ReRF	30.34	0.972	0.055	21.2	0.21	1220	30.33	0.962	0.054	22.4	0.27	843
Ours (low)	30.42	0.966	0.059	0.55	0.1	11.76	27.60	0.950	0.083	0.58	0.12	11.72
Ours (high)	32.57	0.978	0.045			85.33	30.18	0.962	0.056			71.67

Table 1. Results on NHR [42] and ReRF [41] datasets. Training time (T.T. in minutes), Rendering Time (R.T. in seconds), and sizes (in KB) are averaged over frames.

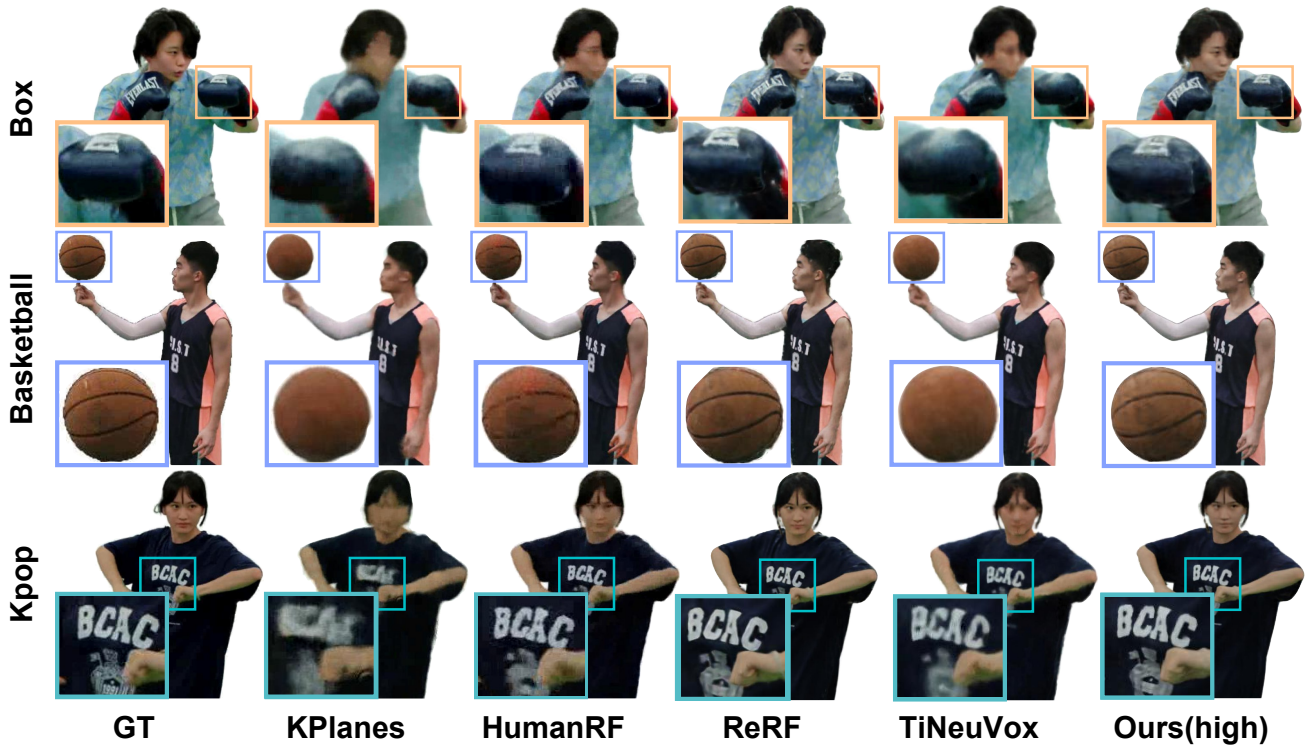

Figure 5. Qualitative comparison. The rendering quality of TeTriRF compared to four baselines KPlanes [9], HumanRF [15], ReRF [41], TiNeuVox [8] on human-centric scenes *box*, *basketball* and *kpop* from NHR and ReRF datasets.

[9], HumanRF [15], TiNeuVox [8], and ReRF [41]. For forward-facing scenes, our approach is compared with KPlanes [9], NeRFPlayer [34], and MixVoxels [38]. For a fair comparison, we use their official codes and align the experimental setups for different datasets. Details can be found in the supplementary materials. We evaluate two versions of TeTriRF: one compressed using a high-quality option in the video codec (denoted as 'Ours(high)' with CRF=20), and another compressed with a lower quality option (denoted as 'Ours(low)' with CRF=33).

Experiment Protocol. For consistency and comparability, we limit the scope of training and evaluation to the initial 200 frames of each scene unless stated otherwise. We specify viewpoints 5 and 41 in the NHR dataset, viewpoints 6 and 39 in the ReRF dataset, and viewpoint 0 in the DyNeRF dataset as the test views, which are excluded during training. All models are benchmarked on an NVIDIA V100 GPU.

Evaluation metrics. Our evaluation framework focuses on three aspects: image quality, running time, and storage. For image quality, we use three standards: Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), and Learned Perceptual Image Patch Similarity (LPIPS) with VGG backbone. We assess storage by calculating the average model size per frame in kilobytes (KB), which is essential for rendering. We measure time efficiency by the average training time per frame in minutes (T.T.) and rendering time per frame in seconds (R.T.). Please note that we

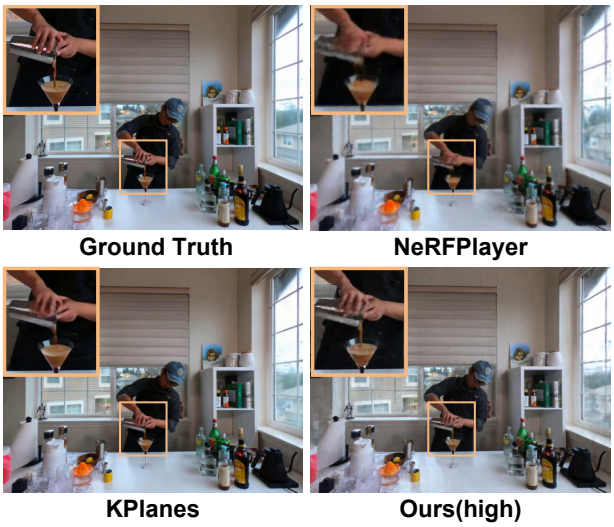

Figure 6. Qualitative results on the forward-facing scene in DyNeRF dataset. Visual comparisons on *coffee martini*. our method can also preserve more details accurately compared to the others, as evident in the clear geometry of fingers and the coffee flow.

evaluate the rendering time in the Python implementation, which includes system overhead. Therefore, it only provides a relative comparison, not the actual rendering time in efficient implementations.

Results. In the assessment of the human-centric dataset detailed in Tab. 1, 'Ours(low)' demonstrates rendering quality that is on par with those of the KPlanes, TiNeuVox. Notably, it achieves this while requiring significantly less storage, over two orders of magnitude lower. Moreover, it offers a rendering time that is at least twice as fast as the most storage efficient SOTA method ReRF [41]. 'Ours(high)' yields further improvements.

Tab. 2 shows the results on the DyNeRF dataset. In this table, we report the rendering quality of NeRFPlayer as it appears in the original paper, and additionally provide test results for running time and storage size, which were obtained by executing the official code. We use MixVoxels-M [38] in our experiments. 'Ours(high)' delivers a rendering quality comparable to that of the other models while achieving significantly better time efficiency and requiring less storage space.

Qualitative comparison between the baselines and our method can be found in Fig. 5 and Fig. 6. Our approach demonstrates the capability in handling intricate details of highly dynamic objects with a reduced model size. For instance, in Fig. 5, our method can effectively capture details of a rapidly spinning basketball.

4.2. Evaluation

Ablation Study We evaluate the progressive scaling (PS) module and group training regularization (Reg) module by disabling them one at a time during training to analyze their individual contribution to the complete TeTriRF model. For this ablation study, we selected the 'sport1' scene from the NHR dataset. In our experiments, we also replace the H265 codec with the MPEG-2 codec to assess how video codecs with different efficiency affect the compression rate and rendering quality. MPEG-2, being an earlier video encoding technology, has simpler algorithms that are akin to those used in ReRF's compression algorithm. By doing this, we aim to draw a comparison between our hybrid representation and the 3D voxel grid utilized in ReRF. Fig. 7 illustrates the rate-distortion curves for various settings. The exclusion of either PG or Reg leads to a deterioration in the performance of TeTriRF. Even when employing a basic video codec like MPEG-2, TeTriRF still manages to outperform ReRF. This suggests that our proposed hybrid representation offers advantages in compressing dynamic scenes.

Fig. 8 presents a qualitative comparison between the complete TeTriRF models at varying sizes and their variants in similar sizes. In the absence of the PS module, the variant generates density floaters around the geometry surfaces, resulting in a marginally blurred RGB image. The variant lacking Reg struggles with geometry reconstruction, primarily due to insufficient temporal consistency. Both ReRF and the variant using MPEG-2 display significant inadequacies at this level of storage.

Storage Breakdown. We analyze the storage components of both 'Ours(low)' and 'Ours(high)', breaking down each component within them. The data is compiled from statistics on 200 frames of the 'sport1' scene. Table 3 shows the results. In the table, we exclude their metadata (e.g. the bounding box), which is less than 1KB..

Long Sequence. We tested the TeTriRF on an ultra-long sequence, specifically the 'Kpop' sequence in the ReRF dataset. Fig. 9 shows the per-frame PSNR curve of TeTriRF

	PSNR	SSIM	LPIPS	T.T.	R.T.	Size
NeRFPlayer	30.293	0.909	0.309	0.25	3.5	2427
KPlanes	31.38	0.940	0.212	0.57	11.5	539
MixVoxels	30.69	0.918	0.236	0.18	0.11	1733
Ours(low)	28.71	0.867	0.321	0.65	0.24	21.46
Ours(high)	30.43	0.906	0.248			62.5

Table 2. Comparison on the forward facing dataset DyNeRF. The training time (T.T. in minutes), rendering time (R.T. in seconds) and model size (Size in KB) are averaged out over the number of frames.

	$\mathbf{P}_{xy}$	$\mathbf{P}_{xz}$	$\mathbf{P}_{yz}$	$\mathbf{V}_{\sigma}$	MLPs
Ours(low)	431	315.2	577.6	558.2	490
Ours(high)	4032	2758.6	5353.8	3350	

Table 3. Analysis of storage components in 'Ours(low)' and 'Ours(high)' based on 200 frames of sport1 scene (Values in KB).

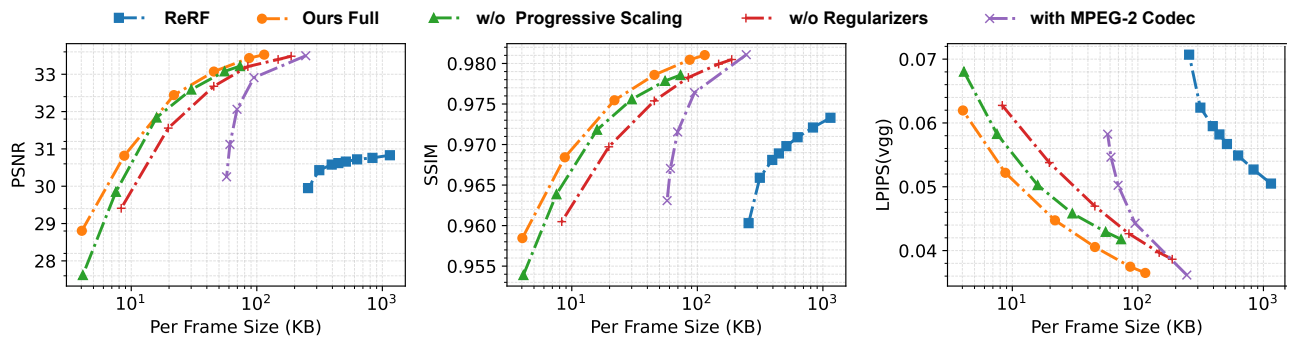

Figure 7. Rate-distortion curves. TeTriRF outperforms alternative versions and ReRF. Disabling progressive scaling (PS) or group-based regularization (Reg) reduces TeTriRF's performance. Even with MPEG-2, TeTriRF excels in compressing dynamic scenes. In the first two line graphs, the closer to the top left corner, the better; in the last one, the bottom left corner is optimal.

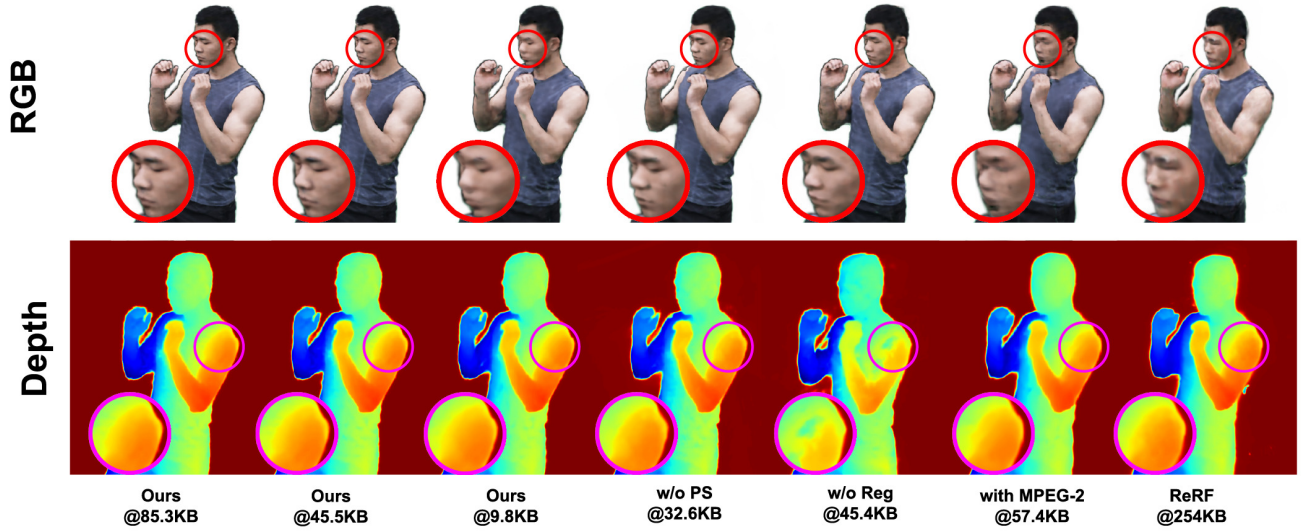

Figure 8. Qualitative results of complete TeTriRF model, its variants and ReRF. The variants are compared at approximately matched sizes.

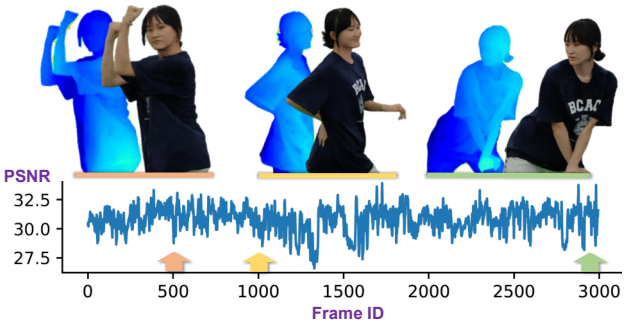

Figure 9. TeTriRF performance on the Kpop sequence, showcasing the first 3000 frames.

over time under 'ours(high)' setting.

5. Conclusion and Future Work

Our proposed innovative training scheme elevates the sequential representation's temporal coherence and low-entropy characteristics, resulting in a dramatic enhancement of compression efficiency. Our evaluation has demonstrated the compactness and effectiveness of TeTriRF's hybrid representation that plays an important part in the compression process. Leveraging our compression pipeline, TeTriRF is able to support long-duration FVV experiences, while remarkably minimizing storage requirements. TeTriRF's rendering pipeline is efficient, straightforward, and opens the door to leveraging GLSL shaders for our method, paving the way for real-time performance on diverse devices, supported by hardware-accelerated video decoding. Training efficiency, fast rendering speed, and compact data storage of TeTriRF enable photo-realistic FVV applications in AR/VR contexts. TeTriRF's rendering speed relies on scene sparsity, sampling only non-empty space. A more efficient pipeline could be beneficial. Currently we tested with stationary cameras, future plans include dynamic scenes from moving cameras.

Acknowledgments. This work was supported by the Flanders AI Research program.

References

[1] 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 (TOG)*, 39(4):86–1, 2020. 2

[2] Ang Cao and Justin Johnson. Hexplane: A fast representation for dynamic scenes. *CVPR*, 2023. 2

[3] Ang Cao and Justin Johnson. Hexplane: A fast representation for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 130–141, 2023. 2

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

[5] Alvaro Collet, Ming Chuang, Pat Sweeney, Don Gillett, Dennis Evseev, David Calabrese, Hugues Hoppe, Adam Kirk, and Steve Sullivan. High-quality streamable free-viewpoint video. *ACM Transactions on Graphics (ToG)*, 34(4):1–13, 2015. 2

[6] Chenxi Lola Deng and Enzo Tartaglione. Compressing explicit voxel grid representations: fast nerfs become also small. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1236–1245, 2023. 3

[7] Yilun Du, Yinan Zhang, Hong-Xing Yu, Joshua B. Tenenbaum, and Jiajun Wu. Neural radiance flow for 4d view synthesis and video processing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. 2

[8] Jiemin Fang, Taoran Yi, Xinggang Wang, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Matthias Nießner, and Qi Tian. Fast dynamic radiance fields with time-aware neural voxels. In *SIGGRAPH Asia 2022 Conference Papers*, 2022. 1, 2, 6

[9] Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12479–12488, 2023. 1, 2, 3, 6

[10] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. Dynamic view synthesis from dynamic monocular video. In *Proceedings of the IEEE International Conference on Computer Vision*, 2021. 2

[11] Haoyu Guo, Sida Peng, Yunzhi Yan, Linzhan Mou, Yujun Shen, Hujun Bao, and Xiaowei Zhou. Compact neural volumetric video representations with dynamic codebooks. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 3

[12] Xiang Guo, Guanying Chen, Yuchao Dai, Xiaoqing Ye, Jiadai Sun, Xiao Tan, and Errui Ding. Neural deformable voxel grid for fast optimization of dynamic view synthesis. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, 2022. 2

[13] Peter Hedman, Pratul P Srinivasan, Ben Mildenhall, Jonathan T Barron, and Paul Debevec. Baking neural radiance fields for real-time view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5875–5884, 2021. 2, 3

[14] Binbin Huang, Xinhao Yan, Anpei Chen, Shenghua Gao, and Jingyi Yu. Pref: Phasorial embedding fields for compact neural representations. *arXiv preprint arXiv:2205.13524*, 2022. 3

[15] Mustafa Işık, Martin Rünz, Markos Georgopoulos, Taras Khakhulin, Jonathan Starck, Lourdes Agapito, and Matthias Nießner. Humanrf: High-fidelity neural radiance fields for humans in motion. *ACM Transactions on Graphics (TOG)*, 42(4):1–12, 2023. 2, 4, 5, 6

[16] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42 (4), 2023. 2

[17] Lingzhi Li, Zhen Shen, Zhongshu Wang, Li Shen, and Liefeng Bo. Compressing volumetric radiance fields to 1 mb. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4222–4231, 2023. 3

[18] Tianye Li, Miroslava Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, S. Lovegrove, Michael Goesele, Richard A. Newcombe, and Zhaoyang Lv. Neural 3d video synthesis from multi-view video. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5511–5521, 2021. 2, 5

[19] Tianye Li, Mira Slavcheva, Michael Zollhöfer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, Richard Newcombe, and Zhaoyang Lv. Neural 3d video synthesis from multi-view video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5521–5531, 2022. 1

[20] Zhong Li, Yu Ji, Wei Yang, Jinwei Ye, and Jingyi Yu. Robust 3d human motion reconstruction via dynamic template construction. In *2017 International Conference on 3D Vision (3DV)*, pages 496–505. IEEE, 2017. 2

[21] Zhong Li, Minye Wu, Wangyiteng Zhou, and Jingyi Yu. 4d human body correspondences from panoramic depth maps. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2877–2886, 2018. 2

[22] Jia-Wei Liu, Yan-Pei Cao, Weijia Mao, Wenqiao Zhang, David Junhao Zhang, Jussi Keppo, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Devrf: Fast deformable voxel radiance fields for dynamic scenes. *Advances in Neural Information Processing Systems*, 35:36762–36775, 2022. 1, 2

[23] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *NeurIPS*, 2020. 2

[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 *ECCV*, 2020. 1, 2, 3

[25] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, 2022. 2

[26] Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Steven M. Seitz, and Ricardo

Martin-Brualla. Nerfies: Deformable neural radiance fields. *ICCV*, 2021. 1, 2

[27] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M. Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *ACM Trans. Graph.*, 40(6), 2021. 2

[28] Sida Peng, Yunzhi Yan, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Representing volumetric videos as dynamic mlp maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4252–4262, 2023. 1, 2

[29] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10318–10327, 2021. 1, 2

[30] Christian Reiser, Rick Szeliski, Dor Verbin, Pratul Srinivasan, Ben Mildenhall, Andreas Geiger, Jon Barron, and Peter Hedman. Merf: Memory-efficient radiance fields for real-time view synthesis in unbounded scenes. *ACM Transactions on Graphics (TOG)*, 42(4):1–12, 2023. 2, 3

[31] Daniel Rho, Byeonghyeon Lee, Seungtae Nam, Joo Chan Lee, Jong Hwan Ko, and Eunbyung Park. Masked wavelet representation for compact neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20680–20690, 2023. 3

[32] Sara Fridovich-Keil and Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *CVPR*, 2022. 2

[33] Ruizhi Shao, Zerong Zheng, Hanzhang Tu, Boning Liu, Hongwen Zhang, and Yebin Liu. Tensor4d: Efficient neural 4d decomposition for high-fidelity dynamic reconstruction and rendering, 2023. 2

[34] Liangchen Song, Anpei Chen, Zhong Li, Zhang Chen, Lele Chen, Junsong Yuan, Yi Xu, and Andreas Geiger. Nerfplayer: A streamable dynamic scene representation with decomposed neural radiance fields. *IEEE Transactions on Visualization and Computer Graphics*, 29(5):2732–2742, 2023. 1, 3, 6

[35] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *CVPR*, 2022. 2, 4, 5

[36] Xin Suo, Yuheng Jiang, Pei Lin, Yingliang Zhang, Minye Wu, Kaiwen Guo, and Lan Xu. Neuralhumanfvv: Real-time neural volumetric human performance rendering using rgb cameras. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6226–6237, 2021. 2

[37] Edgar Tretschk, Ayush Kumar 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. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12939–12950, 2020. 2

[38] Feng Wang, Sinan Tan, Xinghang Li, Zeyue Tian, Yafei Song, and Huaping Liu. Mixed neural voxels for fast multi-view video synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19706–19716, 2023. 6, 7

[39] Liao Wang, Jiakai Zhang, Xinhang Liu, Fuqiang Zhao, Yanshun Zhang, Yingliang Zhang, Minye Wu, Jingyi Yu, and Lan Xu. Fourier plenoctrees for dynamic radiance field rendering in real-time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13524–13534, 2022. 1

[40] Liao Wang, Qiang Hu, Qihan He, Ziyu Wang, Jingyi Yu, Tinne Tuytelaars, Lan Xu, and Minye Wu. Neural residual radiance fields for streamably free-viewpoint videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 76–87, 2023. 1, 4

[41] L. Wang, Q. Hu, Q. He, Z. Wang, J. Yu, T. Tuytelaars, L. Xu, and M. Wu. Neural residual radiance fields for streamably free-viewpoint videos. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 76–87, Los Alamitos, CA, USA, 2023. IEEE Computer Society. 3, 4, 5, 6, 7

[42] Minye Wu, Yuehao Wang, Qiang Hu, and Jingyi Yu. Multi-view neural human rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1682–1691, 2020. 5, 6

[43] W. Xian, J. Huang, J. Kopf, and C. Kim. Space-time neural irradiance fields for free-viewpoint video. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9416–9426, Los Alamitos, CA, USA, 2021. IEEE Computer Society. 2

[44] Zhen Xu, Sida Peng, Haotong Lin, Guangzhao He, Jiaming Sun, Yujun Shen, Hujun Bao, and Xiaowei Zhou. 4k4d: Real-time 4d view synthesis at 4k resolution. *arXiv preprint arXiv:2310.11448*, 2023. 2

[45] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. PlenOctrees for real-time rendering of neural radiance fields. In *ICCV*, 2021. 2

[46] Jiakai Zhang, Xinhang Liu, Xinyi Ye, Fuqiang Zhao, Yanshun Zhang, Minye Wu, Yingliang Zhang, Lan Xu, and Jingyi Yu. Editable free-viewpoint video using a layered neural representation. *ACM Transactions on Graphics (TOG)*, 40:1 – 18, 2021. 2

[47] Fuqiang Zhao, Yuheng Jiang, Kaixin Yao, Jiakai Zhang, Liao Wang, Haizhao Dai, Yuhui Zhong, Yingliang Zhang, Minye Wu, Lan Xu, et al. Human performance modeling and rendering via neural animated mesh. *ACM Transactions on Graphics (TOG)*, 41(6):1–17, 2022. 2