

Im4D: High-Fidelity and Real-Time Novel View Synthesis for Dynamic Scenes

Haotong Lin

Sida Peng*

haotongl@zju.edu.cn

pengsida@zju.edu.cn

State Key Laboratory of CAD&CG,
Zhejiang University
China

Zhen Xu

Tao Xie

Xingyi He

zhenx@zju.edu.cn

taotaoxie@zju.edu.cn

xingyihe@zju.edu.cn

Zhejiang University

China

Hujun Bao

Xiaowei Zhou

bao@cad.zju.edu.cn

xwzhou@zju.edu.cn

State Key Laboratory of CAD&CG,
Zhejiang University
China

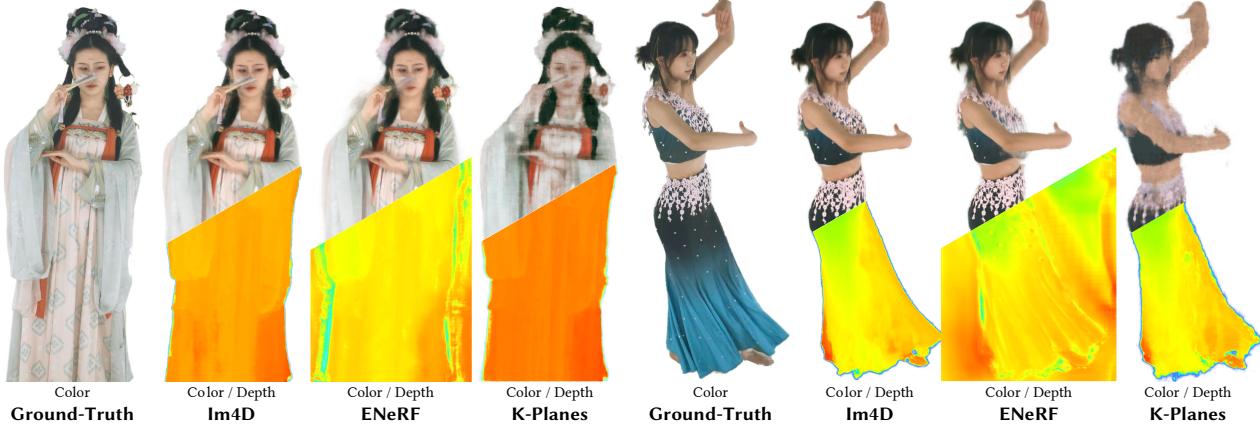


Figure 1: Rendering results on the DNA-Rendering [Cheng et al. 2023] dataset. All the rendered depth maps are normalized using the same method with same hyper parameters and then combined with the rendered alpha maps. Im4D (our method) produces high-fidelity rendering with high-quality depth results on a long dynamic scene with complex motions. ENeRF [Lin et al. 2022] struggles to recover correct depth, leading to flicker and ghosting artifacts. K-Planes [Fridovich-Keil et al. 2023] cannot recover the appearance details on such a difficult scene. Please refer to our video for better visualization.

ABSTRACT

This paper aims to tackle the challenge of dynamic view synthesis from multi-view videos. The key observation is that while previous grid-based methods offer consistent rendering, they fall short in capturing appearance details of a complex dynamic scene, a domain where multi-view image-based rendering methods demonstrate the opposite properties. To combine the best of two worlds, we introduce Im4D, a hybrid scene representation that consists of a grid-based *geometry* representation and a multi-view image-based *appearance* representation. Specifically, the dynamic geometry is encoded as a 4D density function composed of spatiotemporal

feature planes and a small MLP network, which globally models the scene structure and facilitates the rendering consistency. We represent the scene appearance by the original multi-view videos and a network that learns to predict the color of a 3D point from image features, instead of memorizing detailed appearance totally with networks, thereby naturally making the learning of networks easier. Our method is evaluated on five dynamic view synthesis datasets including DyNeRF, ZJU-MoCap, NHR, DNA-Rendering and ENeRF-Outdoor datasets. The results show that Im4D exhibits state-of-the-art performance in rendering quality and can be trained efficiently, while realizing real-time rendering with a speed of 79.8 FPS for 512x512 images, on a single RTX 3090 GPU. The code is available at <https://zju3dv.github.io/im4d>.

CCS CONCEPTS

• Computing methodologies → Image-based rendering.

ACM Reference Format:

Haotong Lin, Sida Peng, Zhen Xu, Tao Xie, Xingyi He, Hujun Bao, and Xiaowei Zhou. 2023. Im4D: High-Fidelity and Real-Time Novel View Synthesis for Dynamic Scenes. In *SIGGRAPH Asia 2023 Conference Papers (SA Conference Papers'23)*, December 12–15, 2023, Sydney, NSW, Australia. ACM, New York, NY, USA, 12 pages. <https://doi.org/10.1145/3610548.3618142>

*Corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SA Conference Papers '23, December 12–15, 2023, Sydney, NSW, Australia

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-9007-7/23/12...\$15.00

<https://doi.org/10.1145/3610548.3618142>

1 INTRODUCTION

Dynamic view synthesis aims to render novel views of a real-world dynamic scene given input videos, which is a long-standing research problem in computer vision and graphics. It has a variety of applications in film and game production, immersive telepresence, sports broadcasting, etc. The key challenge of this problem is to efficiently reconstruct a 4D representation of the dynamic scene from multi-view videos, which allows high-fidelity (i.e., photo-realistic and multi-view consistent) and real-time rendering at arbitrary viewpoints and time.

Recent methods [Mildenhall et al. 2020] have achieved great success in novel view synthesis for static scenes using implicit scene representations, which brings new insights to the field of dynamic view synthesis. Some methods (e.g., [Li et al. 2021b; Xian et al. 2021]) have been proposed to enhance NeRF's [Mildenhall et al. 2020] MLP by incorporating time as an additional input, enabling the representation of radiance fields in dynamic scenes. By employing the volume rendering technique with a series of training strategies, DyNeRF [Li et al. 2021b] is able to achieve realistic rendering after training for one week on 8xV100 GPUs. To address the challenge of training efficiency, recent methods [Fang et al. 2022; Fridovich-Keil et al. 2023] draw inspiration from [Chen et al. 2022b; Müller et al. 2022] and incorporate explicit optimizable embeddings into the implicit representation, significantly accelerating the training speed. However, as depicted in Fig. 1, these methods encounter difficulties in capturing the appearance details of complex dynamic scenes. Some possible solutions are to increase the model size or divide the sequence and process it using multiple models, which will lead to a linear increase in training time and model size.

We observe that 2D videos faithfully record the appearance of scenes and video compression techniques have been well studied and standardized [Sullivan et al. 2012], allowing very efficient storage and transmission. Motivated by this, we propose Im4D, a novel hybrid scene representation for dynamic scenes, which consists of a grid-based 4D geometry representation and a multi-view image-based appearance representation, for efficient training and high-fidelity rendering of complex dynamic scenes. Specifically, given a spatial point at a specific time step, we fetch the corresponding feature from explicit spatiotemporal feature planes [Fridovich-Keil et al. 2023] and then regress the density from this feature with a small MLP. For the appearance part, we first feed the nearby views of the rendered view at the time step into a CNN network to obtain feature maps. Then, we project the spatial point onto these feature maps to obtain pixel-aligned features. Finally, we utilize a small MLP to predict the color from these features. This representation can be rendered using the volume rendering technique. Unlike previous methods [Fang et al. 2022; Fridovich-Keil et al. 2023] that *memorize* the radiance of each space-time point along each direction (in \mathbb{R}^6 space), our method learns *inferring* the radiance from input image features. We experimentally show that the proposed method achieves faster training and better rendering quality, indicating that our strategy effectively reduces the learning burden of the network.

In addition to boosting the training speed and rendering quality, Im4D inherently ensures better cross-view rendering consistency. Previous methods [Lin et al. 2022; Wang et al. 2021] also utilize

an image-based rendering representation, but they simultaneously predict the geometry and appearance for each rendering. Their nature of *per-view* reconstruction instead of *global* reconstruction cannot ensure rendering consistency between viewpoints. With the global geometry representation, Im4D achieves better cross-view rendering consistency than these methods as shown in our video.

We evaluate our method on several commonly used benchmarks for dynamic view synthesis, including the ZJU-MoCap [Peng et al. 2021], NHR [Wu et al. 2020], DyNeRF [Li et al. 2021b] and DNA-Rendering [Cheng et al. 2023] datasets. Our method consistently demonstrates state-of-the-art performance across all of these datasets, achieved with a training time of a few hours. Furthermore, we demonstrate that our representation can be rendered in real-time, capable of rendering 512x512 images at a speed of 79.8 FPS on a single RTX 3090 GPU. We further validate the versatility of our method on the ENeRF-Outdoor [Lin et al. 2022] dataset. This dataset is particularly challenging for dynamic view synthesis, as it comprises large motions within complex outdoor scenes.

In summary, this work makes the following contributions: 1) We propose Im4D, a novel hybrid representation for dynamic scenes, which consists of a grid-based 4D geometry representation and a multi-view image-based appearance representation. 2) We conduct extensive comparisons and ablations on several datasets, demonstrating that our method achieves state-of-the-art performance in terms of rendering quality with training in a few hours. 3) We demonstrate that the proposed representation can be rendered in real-time on the ZJU-MoCap dataset.

2 RELATED WORKS

Novel View Synthesis of Static Scenes. Novel view synthesis has traditionally been approached via several paradigms, including light field-based methods [Davis et al. 2012; Gortler et al. 1996; Levoy and Hanrahan 1996], multi-view images [Buehler et al. 2001; Chaurasia et al. 2013; Flynn et al. 2016; Kalantari et al. 2016; Penner and Zhang 2017; Zitnick et al. 2004], and multi-plane images [Li et al. 2020; Mildenhall et al. 2019; Srinivasan et al. 2019; Szeliski and Golland 1998; Tucker and Snavely 2020; Zhou et al. 2018]. Recently, a new paradigm has emerged in the form of neural representations [Attal et al. 2022; Hedman et al. 2018; Jiang et al. 2020; Kellnhofer et al. 2021; Liu et al. 2019b,a; Lombardi et al. 2019; Shih et al. 2020; Sitzmann et al. 2021, 2019; Suhail et al. 2022; Wizadwongsu et al. 2021] for novel view synthesis. NeRF [Mildenhall et al. 2020] represents scenes as neural radiance fields using a Multi-Layer Perceptron (MLP), yielding impressive rendering results. NeRF requires a lengthy per-scene optimization. To avoid the training burden, several methods [Chen et al. 2021; Chibane et al. 2021; Johari et al. 2022; Liu et al. 2021; Wang et al. 2021; Yu et al. 2021b] propose to use an MLP to decode radiance fields from pixel-aligned image features from nearby images. These methods can be quickly fine-tuned to new scenes by pre-training a CNN to extract feature maps over large datasets. There exist some works [Kopanas et al. 2021; Sun et al. 2020] that study how to select nearby images and develop a smooth fading strategy to avoid temporal instability. Some methods [Chen et al. 2022b; Müller et al. 2022; Sun et al. 2021; Yu et al. 2022] design hybrid or explicit structures to store optimizable features or latent representations (e.g., spherical harmonics), achieving rapid

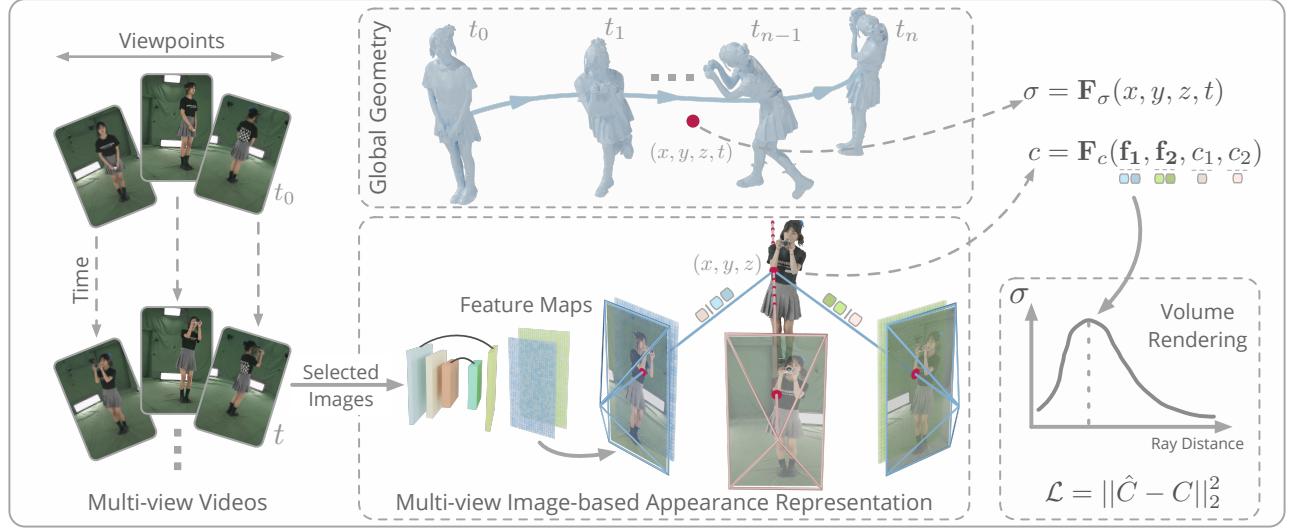


Figure 2: Overview of Im4D. Given a set of multi-view videos, the proposed method aims to reconstruct a 3D model capable of rendering photorealistic images at arbitrary viewpoints and time steps. The proposed method models the geometry with a **global 4D density function**. This function consists of a small MLP and a 4D space structure storing optimizable features. The appearance part is represented with a multi-view image-based appearance model, which learns to predict the color of a 3D point from image features extracted from selected images (the input images closest to the rendering view).

reconstruction. To improve the rendering speed, some methods [Garbin et al. 2021; Hedman et al. 2021; Liu et al. 2020; Reiser et al. 2021; Yu et al. 2021a] represent or cache the radiance fields into efficient structures. Another category of methods [Neff et al. 2021] has also achieved significant acceleration by using depth to expedite rendering. More recently, several methods [Chen et al. 2022a; Wan et al. 2023] design representations based on polygonal mesh and leverage graphics pipelines to achieve real-time rendering effects. A few studies [Barron et al. 2021, 2022, 2023; Tancik et al. 2022; Turki et al. 2022] have also analyzed issues such as anti-aliasing inherent in NeRF representation. In addition to NeRF-based methods, several methods [Riegler and Koltun 2020, 2021] have improved rendering quality or reconstruction quality by incorporating multi-view image-based rendering appearance models into surface models. [Bergman et al. 2021] explores a similar idea that uses multi-view images and an MLP for appearance and geometry, respectively.

Novel View Synthesis of Dynamic Scenes. Early works [Dou et al. 2016; Newcombe et al. 2015; Orts-Escalano et al. 2016; Yu et al. 2018] predominantly used explicit surface models for dynamic scenes. They [Collet et al. 2015] rely on depth sensors and multi-view stereo techniques to capture per-view depth before consolidating it into the scene geometry. In a different approach, NeuralVolumes [Lombardi et al. 2019] use volume rendering to reconstruct 4D scenes from color images. However, its usage of 3D volumes imposed limitations on achieving high-resolution results. In response, some works [Lombardi et al. 2021; Peng et al. 2023] seek to convert the 3D volume into a 2D representation, as MLP maps and UV maps, respectively. While they provide a lightweight solution that is capable of real-time rendering, they also demand extended training periods. Progressing further, a collection of studies extended NeRF to accommodate dynamic scenes, paving the way for high-resolution

rendering. These methods generally incorporated a time variable into the NeRF's MLP [Li et al. 2021b; Lin et al. 2023; Xian et al. 2021], scene flows [Du et al. 2021; Gao et al. 2021; Li et al. 2021a; You and Hou 2023] or use deformable fields [Park et al. 2021a,b; Pumarola et al. 2021; Zhang et al. 2021]. Despite the promising results, these techniques necessitated considerable training resources. To optimize training efficiency, recent methodologies [Attal et al. 2023; Cao and Johnson 2023; Fang et al. 2022; Fridovich-Keil et al. 2023; Gan et al. 2022; Shao et al. 2022; Wang et al. 2022] introduced optimizable embeddings into the implicit representation, resulting in training speed-ups. However, as demonstrated by our experiments, K-Planes [Fridovich-Keil et al. 2023] still falls short of delivering clear rendering results on challenging dynamic scenes. Some recent methods [Li et al. 2022a; Song et al. 2023; Wang et al. 2023] also explore the streaming representation of dynamic scenes. Another line of works such as [Lin et al. 2022; Wang et al. 2021] deploy multi-view images to represent 4D scenes, offering high-quality rendering and efficient training. Nonetheless, due to their inherent per-view reconstruction nature, they are unable to guarantee cross-view rendering consistency. In contrast to these, our hybrid dynamic scene representation ensures rendering consistency by employing a global grid-based geometry representation. At the same time, we use multi-view images for appearance representation, achieving high-quality rendering. More recently, a concurrent work [İşik et al. 2023] has similar findings that the grid-based representations significantly degrade when dealing with long and complex dynamic scenes. They address this issue by adaptively dividing the dynamic scene into much shorter segments.

3 METHOD

Given multi-view videos of a dynamic scene, our objective is to construct a time-varying 3D model capable of generating photorealistic

images from any perspective and any time (discrete time steps of input frames). We propose Im4D, a novel scene representation, which combines the strengths of both *global* methods [Fridovich-Keil et al. 2023; Mildenhall et al. 2020], with their cross-view rendering consistency, and multi-view image-based rendering (*per-view*) methods [Wang et al. 2021], recognized for their high-quality rendering and fast training [Lin et al. 2022]. Im4D consists of two parts: a global grid-based dynamic geometry representation (Sec. 3.1) and a multi-view image-based appearance representation (Sec. 3.2). As shown in Fig. 2, we represent the density fields with a continuous function that takes (x, y, z, t) as input. The appearance is modeled using a multi-view image-based rendering model, which infers the color of a 3D point from nearby multi-view images at the given time. The proposed representation is optimized using RGB images and can be rendered in real-time. (Sec. 3.3).

3.1 Grid-based Dynamic Geometry

In this section, we seek to represent the dynamic geometry of a scene as a time-varying 3D model to achieve inter-view rendering consistency. Inspired by recent progress in static 3D reconstruction [Mildenhall et al. 2020; Müller et al. 2022; Yu et al. 2022], we represent the dynamic geometry of a scene as a grid-based model.

In static 3D reconstruction, the geometry of a scene is represented as a 3D vector-valued function that takes a 3D position coordinate as input and outputs a volume density. To consistently represent the dynamic geometry, we extend the static geometry function to a 4D vector-valued function, where the input becomes a 3D coordinate (x, y, z) and time t :

$$\sigma = F_\sigma(x, y, z, t). \quad (1)$$

In practice, we utilize a hybrid representation to implement F_σ , which interpolates the (x, y, z, t) in the 4D volume V to obtain $v(x, y, z, t)$ and then a small MLP network m_σ maps the feature $v(x, y, z, t)$ to a scalar density σ . Inspired by [Chan et al. 2022; Chen et al. 2022b], we decompose the 4D volume V into six orthogonal planes $\{P_i | i \in xy, xz, yz, xt, yt, zt\}$ to maintain efficiency in storage. Thus the feature $v(x, y, z, t)$ can be defined as the aggregation of features $\{p_i = \text{interp}(P_i, i) | i \in xy, xz, yz, xt, yt, zt\}$. For simplicity, we simply use concatenation as the aggregation function. The final dynamic geometry model can be formulated as:

$$\sigma = m_\sigma(p_{xy} \oplus p_{xz} \oplus p_{yz} \oplus p_{xt} \oplus p_{yt} \oplus p_{zt}). \quad (2)$$

The proposed grid-based dynamic geometry function is global and continuous in both space and time. With the volume rendering technique, the proposed model inherently ensures cross-view rendering consistency. Rather than representing the geometry as a global function, previous multi-view image-based methods [Lin et al. 2022; Wang et al. 2021] predict the geometry for a novel viewpoint from the nearby multi-view images. This can be regarded as *per-view* reconstruction instead of the *global* reconstruction, which makes it difficult to achieve consistent rendering, resulting in flickering artifacts. Some concurrent works [Cao and Johnson 2023; Fridovich-Keil et al. 2023; Shao et al. 2022] propose similar representations to represent the dynamic scene. However, they represent both geometry and appearance as a global function, which struggles to achieve high-fidelity rendering. In contrast, we only represent

the geometry as a global 4D function. Next, we will introduce how to efficiently represent the appearance of a dynamic scene.

3.2 Multi-view Image-based Appearance

This section delves into finding a representation for high-fidelity appearance model. Specifically, we aim to predict the radiance of the space-time point (x, y, z, t) along the view direction d . A straightforward approach would be to extend Eq. 1 to predict an additional geometry feature, and then use an MLP that takes the geometry feature and view direction d as inputs to predict radiance. This method is commonplace in static scenes [Mildenhall et al. 2020; Müller et al. 2022]. However, the same strategy, when applied to complex dynamic scenes where the information becomes more abundant and complex, tends to render blurry results as shown in Fig. 1. To address this issue, we propose a multi-view image-based appearance model. Rather than modeling the appearance as an MLP with 4D explicit structure, we predict the color from the image features, which are projected from multi-view image feature maps of the 4D space-time point. In this manner, the appearance model only needs to learn the problem of *inferring*, rather than *memorizing* the radiance values at each point along each viewpoint in space and time (in \mathbb{R}^6 space).

Specifically, to render an image from a novel space-time viewpoint, we first select N_v input images from the input multiple videos that are spatially close to the desired viewpoint at the given time. The spatial proximity is defined as the distance between camera positions. Then we use a 2D UNet [Ronneberger et al. 2015] to extract features from the N_v selected images, resulting in $\{M_i | i \in [0, N_v]\}$, where $M_i \in \mathbb{R}^{C_i \times H_i \times W_i}$ is the feature map extracted from the i -th view, and C_i, H_i, W_i denote the number of channels, height, and width of the feature map, respectively. Next, we project the 3D point (x, y, z) to the N_v views to obtain the corresponding 2D coordinates $\{u_i | i \in [0, N_v]\}$, where $u_i \in \mathbb{R}^2$ is the projected 2D coordinates of the i -th view. Then we use the bilinear sampling to sample the feature map M_i at the projected coordinates u_i to obtain the projected features $\{f_i | i \in [0, N_v]\}$ and pixel colors $\{c_i | i \in [0, N_v]\}$. The projected features $\{f_i | i \in [0, N_v]\}$ are aggregated to obtain the fused feature f . For simplicity, we use variance and mean as the aggregation function, which is formulated as:

$$f = \text{VAR}(\{f_i | i \in [0, N_v]\}) \oplus \text{MEAN}(\{f_i | i \in [0, N_v]\}). \quad (3)$$

The color c of point (x, y, z, t) is predicted using a pointnet-like [Qi et al. 2017] structure similar to ENERF [Lin et al. 2022], which is invariant to the order of the input features:

$$c = \frac{\sum_i^{N_v} m_c(f, f_i, \text{diff}(d, d_i)) * c_i}{\sum_i^{N_v} m_c(f, f_i, \text{diff}(d, d_i))}, \quad (4)$$

where d_i is the view direction from i -th view camera position to point (x, y, z) and $\text{diff}(d, d_i)$ is the normalized angle difference between d and d_i .

3.3 Efficient Training and Rendering

Optimization. In the previous sections, we described how to obtain color and view-dependent radiance using the proposed scene representation. Given the estimated density and radiance fields, we employ the volume rendering technique [Mildenhall et al. 2020] to

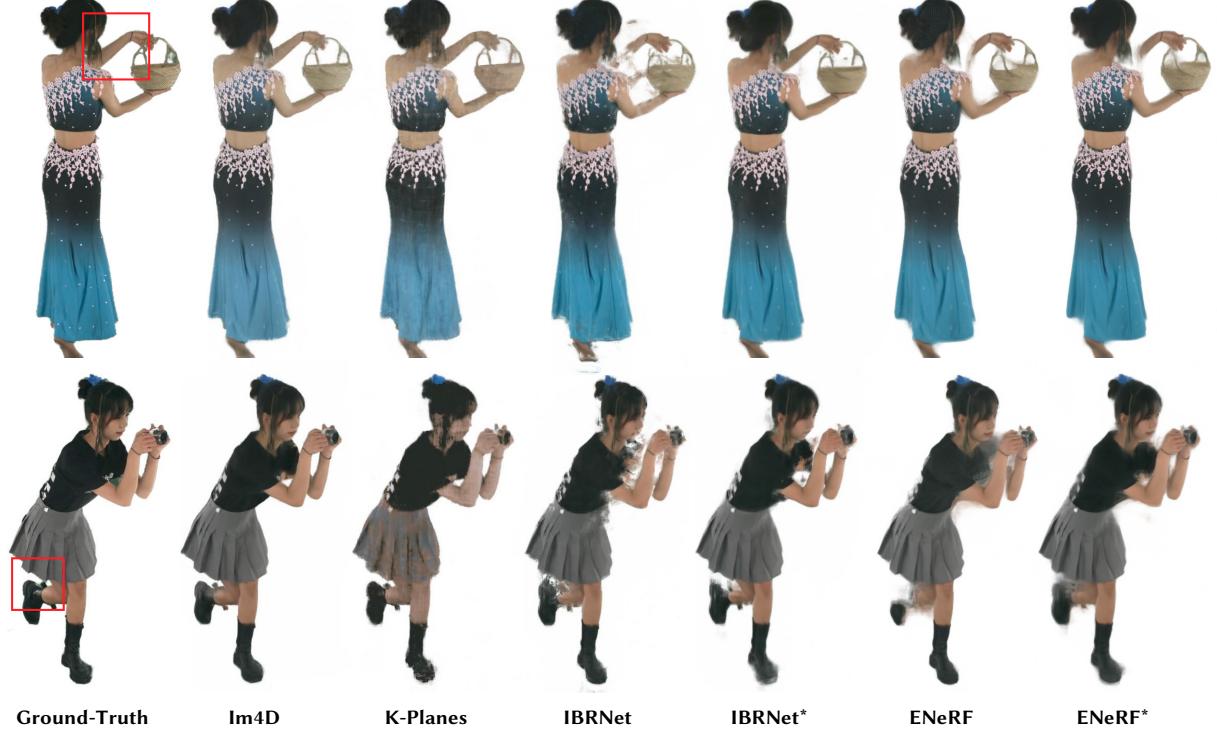


Figure 3: Qualitative comparison of image synthesis results on the DNA-Rendering dataset. The superscript * implies that the results are obtained with extensive per-scene fine-tuning. IBRNet and ENeRF often produce artifacts in thin structures or occluded regions. Our method produces high-fidelity rendering and superior results in these regions, owing to the global geometry representation. K-Planes struggles to recover the appearance details.

render pixel colors. We optimize our scene representation using the mean squared error (MSE) loss with color images. In addition to the color loss, we follow [Fridovich-Keil et al. 2023; Yu et al. 2022] and use the total variation (TV) loss to regularize the explicit feature planes $\{\mathbf{P}_i | i \in \{xy, xz, yz, xt, yt, zt\}\}$. We detail our loss formulation in the supplementary material.

Efficient Training. In Table 3, our experiments show that our approach with an image-based appearance model obtains higher rendering quality at 100K iterations than the one without image-based appearance model training with 480K iterations. To further improve the training speed, we exploit the property of a multi-view image-based appearance model and design a specialized training strategy. The basic idea is that the appearance model can be quickly trained as it is an inferring model. We disable the training of the appearance model to avoid the backward time, which accelerates one training iteration. Specifically, we first jointly train the image feature network with the geometry model for N_j iterations and then finetune the image feature network every N_f iterations. In practice, we set N_j to 5000 and N_f to 20 in all experiments.

Efficient Rendering. Previous multi-view image-based rendering methods, such as ENeRF [Lin et al. 2022], demonstrate that estimating the depth of novel views as coarse geometry to guide sampling can accelerate rendering. In our approach, we have a global geometry model, which enables us to accelerate rendering

by precomputing the global coarse geometry. Specifically, we compute a binary field that indicates whether each voxel is occupied or empty. This binary field is then used to guide sampling, allowing us to skip sampling in empty regions and thus speeding up the rendering process. We use NerfAcc [Li et al. 2023] to implement the described efficient rendering strategy. The storage requirement for this binary field is remarkably small. For example, storing a binary field for 300 frames of size 64x64x128 in the ZJU-MoCap dataset only requires 18.75MB, and can be losslessly compressed further to 1.1MB. This acceleration technique can be more efficient than ENeRF as we only need to compute the coarse geometry (the *global* binary field) for once, while ENeRF needs to estimate the coarse geometry (*per-view* depth) for each rendering. (Ours 79 FPS v.s. ENeRF 51FPS for rendering images of 512x512 on the ZJU-MoCap.)

4 EXPERIMENTS

4.1 Datasets and Metrics

We evaluate our method on 4 datasets for dynamic view synthesis, including DNA-Rendering [Cheng et al. 2023], ZJU-MoCap [Peng et al. 2021], NHR [Wu et al. 2020], and DyNeRF [Li et al. 2021b]. DNA-Rendering contains dynamic objects and humans. Its videos are recorded at 15 FPS and each clip lasts for 10 seconds. DNA-Rendering is very challenging due to the complex clothing textures and movements featured in the videos. The NHR dataset features a frame rate of 30FPS over a 3.5-second duration, while

Table 1: Quantitative comparison on the DNA-Rendering dataset. The zero training time means that methods are tested with their released pre-trained model. We observe that our method typically converges after 140k training iterations and report the corresponding training time. We additionally report our results at the 20k iterations to show the training efficiency. For other methods, we identify a specific training step to ensure convergence, and then report the training time for these methods at this fixed training step. (120k and 480k training iterations for K-Planes). The best and second-best results are highlighted green and yellow, respectively.

	Training time (hour)	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow
K-Planes	2	26.34	0.943	0.134
	8	27.45	0.952	0.118
IBRNet	0	25.31	0.954	0.106
	3.5	27.85	0.967	0.081
ENeRF	0	26.85	0.966	0.073
	3.5	28.07	0.968	0.066
Im4D	0.51	27.14	0.961	0.083
	3.33	28.99	0.973	0.062

the ZJU-MoCap dataset runs at 50FPS over 6 seconds. We evaluate our method on 4 sequences for both NHR and DNA-Rendering datasets with all frames, using 90% of the views for training and the remaining views for evaluation. We conduct experiments on 9 sequences for the ZJU-MoCap dataset. The DyNeRF dataset contains dynamic foreground objects and complex background scenes. It is captured by 15-20 cameras of 30FPS@10 seconds. We conduct quantitative experiments on one sequence, using one view as the test set and the remaining views as training data. All images are resized to a ratio of 0.5 for training and testing. We also include the view synthesis results of our method on the ENeRF-Outdoor dataset [Lin et al. 2022] in the supplementary video.

To evaluate the dynamic view synthesis task, NeuralBody [Peng et al. 2021] suggests evaluating metrics only for the dynamic regions. For MoCap datasets, dynamic regions can be obtained by projecting a predefined 3D bounding box of the person onto the images. We follow this definition for the NHR, ZJU-MoCap, and DNA-Rendering datasets. For the DyNeRF dataset, previous methods [Fridovich-Keil et al. 2023; Li et al. 2021b] directly evaluated the entire image. However, we argue that this approach is unreasonable due to the very small motion in this dataset. We introduce a new evaluation approach to specifically evaluate the dynamic regions. Specifically, for a 10-second video, we take the first frame of each second as test frames. For each test frame, we calculate the average frame for that second. Then, we identify several non-overlapping patches with the highest average pixel differences between the test frame and the average image. Only these patches are used for evaluation. We recommend taking 6 patches with a patch size of 128 for the DyNeRF dataset.

4.2 Comparisons with State-of-the-art Methods

Comparison methods. We make comparisons with several open-sourced SOTA methods. These methods can be divided into two categories: 1) methods that optimize *per-view* radiance fields (multi-view image-based methods), including IBRNet [Wang et al. 2021]

Table 2: Quantitative comparison on the DyNeRF dataset. We include quantitative results for both entire image and dynamic regions (entire/dynamic). The description of this evaluation setting can be found in Sec. 4.1.

	Training time	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow
K-Planes	0.8	30.78/27.29	0.953/0.887	0.218/0.379
	2	31.61/29.62	0.961/0.916	0.182/0.306
IBRNet	3	31.52/31.91	0.963/0.956	0.169/0.144
Im4D	0.46	32.58/32.05	0.971/0.956	0.208/0.170

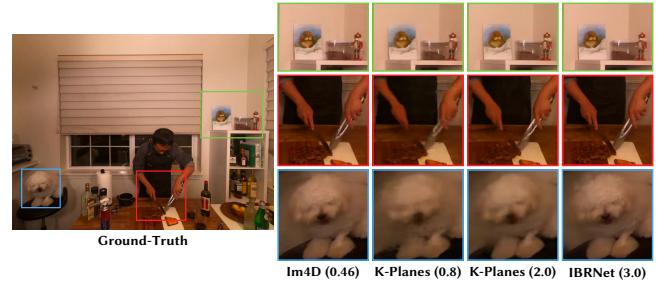


Figure 4: Qualitative comparison on the DyNeRF dataset.

and ENeRF [Lin et al. 2022]. 2) methods that optimize *global* scene representations, including DyNeRF [Li et al. 2022b], K-Planes [Fridovich-Keil et al. 2023] and MLP-Maps [Peng et al. 2023]. We use the official implementation of them. Our comparison setting strictly follows MLP-Maps and some quantitative results are taken from it.

Comparison results. We first make comparisons with the state-of-the-art methods on the DNA-Rendering dataset. The corresponding qualitative and quantitative results are shown in Fig. 3 and Table 1, respectively. As shown in the results, K-Planes struggles to recover the appearance details even after training for eight hours. Other image-based methods (IBRNet, ENeRF and Our method) can recover the high-quality appearance, but IBRNet and ENeRF have artifacts in the occluded regions and regions including thin structures (e.g., legs and hands). We include more comparison results in the supplementary video, where the artifacts for ENeRF and IBRNet can be observed more clearly. We also make quantitative comparisons with the state-of-the-art methods on the ZJU-MoCap and NHR datasets. As shown in Table 3, our method outperforms all the other methods in terms of all metrics on both datasets. Table 2 and Fig. 4 provide quantitative and qualitative comparison results on the DyNeRF dataset, respectively. We include the per-scene breakdown in the supplementary material.

4.3 Ablations and Analysis

Ablations. We conduct qualitative and quantitative ablation studies on 2 sequences from NHR and DNA-Rendering datasets. As shown in Table 4, we first investigate the core components of our approach, i.e., the roles of the multi-view image-based appearance model and grid-based global geometry. 2) is analogous to K-Planes while 3) resembles IBRNet without the ray transformer proposed in IBRNet. Quantitative results demonstrate that the absence of the multi-view image-based appearance model and global geometry

Table 3: Quantitative comparison on ZJU-MoCap and NHR datasets. The results of DyNeRF* and MLP-Maps* are taken from MLP-Maps. The zero training time means that the method is tested with their released pre-trained models. Please refer to the supplementary for qualitative results.

	Training time (hour)	ZJU-MoCap			NHR		
		PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
DyNeRF*	>24	29.88	0.959	0.087	30.87	0.943	0.118
MLP-Maps*	>24	30.17	0.963	0.068	32.20	0.953	0.080
K-Planes	2	29.50	0.956	0.107	30.41	0.943	0.137
	8	30.16	0.962	0.082	32.93	0.958	0.101
IBRNet	0	27.94	0.935	0.126	28.63	0.935	0.113
	2.5	29.40	0.956	0.084	33.53	0.965	0.077
ENeRF	0	29.10	0.959	0.051	26.39	0.931	0.088
	2.5	29.21	0.959	0.049	30.56	0.954	0.074
Im4D	0.49	30.07	0.964	0.061	32.40	0.962	0.074
	2.29	30.49	0.966	0.049	33.72	0.970	0.055

would lead to extremely poor performance. This becomes more evident from the Fig. 5, where the lack of the multi-view image-based appearance model results in lost image detail, and the absence of grid-based *global* geometry induces numerous floater artifacts. Furthermore, we explore the effects of other components of our method, including the proposed efficient training strategy. Initially, we find that not training the image-based appearance model, as in 4), results in poor performance. Not employing our specialized training strategy, as in 7), leads to more extended training periods (approximately an extra hour). Subsequently, we discover that missing the joint training ($N_j = 0$) or not finetuning the image-feature network ($N_f = \infty$) would result in slightly inferior outcomes.

Rendering time analysis. Our rendering time consists of the time for the images to pass through the feature network and the time for volume rendering each ray. On the ZJU-MoCap dataset, the time for the images to pass through the feature network is 1.26ms, and the time for rendering 512x512 rays is 11.27ms. Without using the acceleration strategy, rendering 512x512 rays takes 449.1ms. The final rendering times are 79.8FPS and 2.22FPS, respectively. Note that after using the acceleration strategy, the rendering quality of our method has slightly decreased (0.09 in PSNR, 0.0004 in LPIPS), which we believe is acceptable.

Storage analysis. The storage of the proposed representation consists of the storage for the feature network, the spatial-temporal feature grids, the small MLPs, the binary field (for efficient rendering), and multiple videos. On the NHR dataset, the storage for the feature network is 161KB, the storage for the spatial-temporal feature grids is 82MB, the storage for the small MLPs is 154KB, the binary field is 2.2MB, and the storage for multiple (52 for NHR) lossless (pngs archived with zip format) image sequences is 300.85MB. By using the video compression techniques referenced in [Sullivan et al. 2012], the image sequences can be compressed to 11.14MB with a PSNR error within 0.2 and an LPIPS error within 0.002.

Table 4: Quantitative ablation study on NHR and DNA-Rendering datasets. “Tt” represents the training time, and its corresponding unit is hours. All the models are trained with the same training iterations. Please refer to Sec. 4.3 for the detailed description.

		NHR			DNA-Rendering		
		PSNR ↑	LPIPS ↓	Tt ↓	PSNR ↑	LPIPS ↓	Tt ↓
1) Complete model		34.87	0.043	2.29	29.82	0.045	3.33
2) w/o image-based appearance		31.79	0.107	1.81	26.07	0.102	2.54
3) w/o global geometry		24.71	0.152	2.81	27.44	0.100	4.04
4) $N_j = 0$, $N_f = \infty$		30.44	0.051	2.21	28.25	0.047	3.23
5) $N_j = 0$, $N_f = 20$		34.30	0.044	2.25	29.63	0.047	3.29
6) $N_j = 5000$, $N_f = \infty$		34.29	0.043	2.26	29.27	0.045	3.27
7) $N_j = \infty$, $N_f = \setminus$		34.89	0.042	3.19	29.83	0.045	4.55

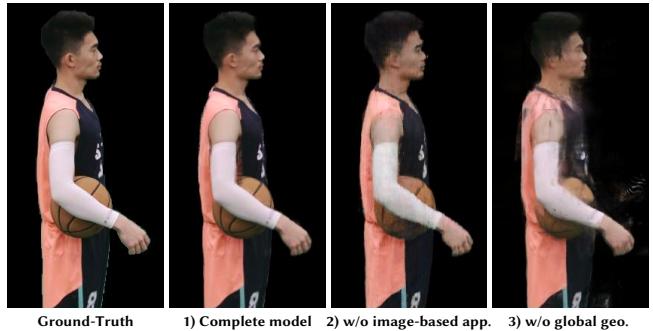


Figure 5: Qualitative ablation study on the NHR dataset. “app.” and “geo.” denote appearance and geometry, respectively.

5 CONCLUSION AND DISCUSSION

This paper introduced Im4D, a novel hybrid scene representation for dynamic scenes that consists of a grid-based 4D geometry representation and a multi-view image-based appearance representation. The proposed method consistently demonstrates superior performance across various benchmarks for dynamic view synthesis including DyNeRF, NHR, ZJU-MoCap and DNA-Rendering datasets, achieving state-of-the-art rendering quality within a few hours of training. Furthermore, the proposed scene representation can be rendered in real-time and handle complex motions within a diverse set of scenarios including outdoor challenging scenes as shown in our video, demonstrating its robustness.

This work still has some limitations. Although the proposed method significantly reduces the rendering artifacts of previous multi-view image-based rendering methods, it has the natural limitation that some regions may be occluded in the input views, which may lead to incorrect appearance prediction. Future work could consider leveraging our consistent dynamic geometry for improved occlusion handling and source view selection to address this issue. Another limitation is that our method cannot handle the monocular video as input, since our appearance representation requires multiple input views at the same moment.

ACKNOWLEDGMENTS

The authors would like to acknowledge support from NSFC (No. 62172364), Information Technology Center and State Key Lab of CAD&CG, Zhejiang University.

REFERENCES

- Benjamin Attal, Jia-Bin Huang, Christian Richardt, Michael Zollhoefer, Johannes Kopf, Matthew O'Toole, and Changil Kim. 2023. HyperReel: High-Fidelity 6-DoF Video with Ray-Conditioned Sampling. *arXiv preprint arXiv:2301.02238* (2023).
- Benjamin Attal, Jia-Bin Huang, Michael Zollhoefer, Johannes Kopf, and Changil Kim. 2022. Learning neural light fields with ray-space embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 19819–19829.
- Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. 2021. Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields. *arXiv* (2021).
- Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. 2022. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5470–5479.
- Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. 2023. Zip-NeRF: Anti-Aliased Grid-Based Neural Radiance Fields. *arXiv preprint arXiv:2304.06706* (2023).
- Alexander Bergman, Petr Kellnhofer, and Gordon Wetzstein. 2021. Fast training of neural lumigraph representations using meta learning. *Advances in Neural Information Processing Systems* 34 (2021), 172–186.
- 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 (SIGGRAPH '01)*. Association for Computing Machinery, New York, NY, USA, 425–432. <https://doi.org/10.1145/383259.383309>
- Ang Cao and Justin Johnson. 2023. Hexplane: A fast representation for dynamic scenes. *arXiv* (2023).
- Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. 2022. Efficient geometry-aware 3D generative adversarial networks. In *CVPR*.
- Gaurav Chaurasia, Sylvain Duchene, Olga Sorkine-Hornung, and George Drettakis. 2013. Depth synthesis and local warps for plausible image-based navigation. *ACM TOG* (2013).
- Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. 2022b. TensoRF: Tensorial Radiance Fields. *arXiv* (2022).
- Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. 2021. MVSNeRF: Fast Generalizable Radiance Field Reconstruction From Multi-View Stereo. In *ICCV*.
- Zhiqin Chen, Thomas Funkhouser, Peter Hedman, and Andrea Tagliasacchi. 2022a. Mobilenerf: Exploiting the polygon rasterization pipeline for efficient neural field rendering on mobile architectures. *arXiv preprint arXiv:2208.00277* (2022).
- Wei Cheng, Ruixiang Chen, Wanqi Yin, Siming Fan, Keyu Chen, Honglin He, Huiwen Luo, Zhongang Cai, Jingbo Wang, Yang Gao, Zhengming Yu, Zhengyu Lin, Daxuan Ren, Lei Yang, Ziwei Liu, Chen Change Loy, Chen Qian, Wayne Wu, Dahua Lin, Bo Dai, and Kwan-Yee Lin. 2023. DNA-Rendering: A Diverse Neural Actor Repository for High-Fidelity Human-centric Rendering. *ICCV* (2023).
- Julian Chibane, Aayush Bansal, Verica Lazova, and Gerard Pons-Moll. 2021. Stereo Radiance Fields (SRF): Learning View Synthesis for Sparse Views of Novel Scenes. In *CVPR*.
- Alvaro Collet, Ming Chuang, Pat Sweeney, Don Gillett, Dennis Evseev, David Calabrese, Hugues Hoppe, Adam Kirk, and Steve Sullivan. 2015. High-quality streamable free-viewpoint video. *ACM TOG* (2015).
- Abe Davis, Marc Levoy, and Fredo Durand. 2012. Unstructured light fields. In *Eurographics*.
- Mingsong Dou, Sameh Khamis, Yury Degtyarev, Philip Davidson, Sean Ryan Fanello, Adarsh Kowdle, Sergio Orts Escolano, Christoph Rhemann, David Kim, Jonathan Taylor, et al. 2016. Fusion4d: Real-time performance capture of challenging scenes. *ACM TOG* (2016).
- Yilun Du, Yinan Zhang, Hong-Xing Yu, Joshua B. Tenenbaum, and Jiajun Wu. 2021. Neural Radiance Flow for 4D View Synthesis and Video Processing. In *ICCV*.
- Jiemin Fang, Taoran Yi, Xinggang Wang, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Matthias Nießner, and Qi Tian. 2022. Fast dynamic radiance fields with time-aware neural voxels. In *SIGGRAPH Asia 2022 Conference Papers*. 1–9.
- John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. 2016. DeepStereo: Learning to Predict New Views From the World's Imagery. In *CVPR*.
- Sara Fridovich-Keil, Giacomo Meanti, Frederik Warburg, Benjamin Recht, and Angjoo Kanazawa. 2023. K-planes: Explicit radiance fields in space, time, and appearance. *arXiv* (2023).
- Wanshui Gan, Hongbin Xu, Yi Huang, Shifeng Chen, and Naoto Yokoya. 2022. V4d: Voxel for 4d novel view synthesis. *arXiv preprint arXiv:2205.14332* (2022).
- Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. 2021. Dynamic view synthesis from dynamic monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 5712–5721.
- Stephan J. Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. 2021. FastNeRF: High-Fidelity Neural Rendering at 200FPS. In *ICCV*.
- Steven J Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F Cohen. 1996. The lumigraph. In *SIGGRAPH*.
- Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. 2018. Deep blending for free-viewpoint image-based rendering. *ACM TOG* (2018).
- Peter Hedman, Pratul P. Srinivasan, Ben Mildenhall, Jonathan T. Barron, and Paul Debevec. 2021. Baking Neural Radiance Fields for Real-Time View Synthesis. In *ICCV*.
- Mustafa İşık, Martin Rünz, Markos Georgopoulos, Taras Khakhulin, Jonathan Starck, Lourdes Agapito, and Matthias Nießner. 2023. HumanRF: High-Fidelity Neural Radiance Fields for Humans in Motion. *arXiv preprint arXiv:2305.06356* (2023).
- Yue Jiang, Dantong Ji, Zhizhong Han, and Matthias Zwicker. 2020. Sdfdiff: Differentiable rendering of signed distance fields for 3d shape optimization. In *CVPR*.
- Mohammad Mahdi Johari, Yann Lepoittevin, and François Fleuret. 2022. GeoNeRF: Generalizing NeRF with Geometry Priors. *CVPR* (2022).
- Nima Khademi Kalantari, Ting-Chun Wang, and Ravi Ramamoorthi. 2016. Learning-based view synthesis for light field cameras. *ACM TOG* (2016).
- Petr Kellnhofer, Lars C Jebe, Andrew Jones, Ryan Spicer, Kari Pulli, and Gordon Wetzstein. 2021. Neural lumigraph rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4287–4297.
- Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*. <http://arxiv.org/abs/1412.6980>
- Georgios Kapanas, Julien Philip, Thomas Leimkühler, and George Drettakis. 2021. Point-Based Neural Rendering with Per-View Optimization. In *Computer Graphics Forum*, Vol. 40. Wiley Online Library, 29–43.
- Marc Levoy and Pat Hanrahan. 1996. Light field rendering. In *SIGGRAPH*.
- Lingzhi Li, Zhen Shen, Li Shen, Ping Tan, et al. 2022a. Streaming Radiance Fields for 3D Video Synthesis. In *Advances in Neural Information Processing Systems*.
- Ruilong Li, Hang Gao, Matthew Tancik, and Angjoo Kanazawa. 2023. NerfAcc: Efficient Sampling Accelerates NeRFs. *arXiv preprint arXiv:2305.04966* (2023).
- Tianye Li, Mira Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, and Zhaoyang Lv. 2021b. Neural 3d video synthesis. *arXiv preprint arXiv:2103.02597* (2021).
- Tianye Li, Mira Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, and Zhaoyang Lv. 2022b. Neural 3d video synthesis. *CVPR* (2022).
- Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. 2021a. Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes. In *CVPR*.
- Zhengqi Li, Wenqi Xian, Abe Davis, and Noah Snavely. 2020. Crowdsampling the plenoptic function. In *ECCV*.
- Haotong Lin, Sida Peng, Zhen Xu, Yunzhi Yan, Qing Shuai, Hujun Bao, and Xiaowei Zhou. 2022. Efficient Neural Radiance Fields for Interactive Free-viewpoint Video. In *SIGGRAPH Asia Conference Proceedings*.
- Haotong Lin, Qianqian Wang, Ruojin Cai, Sida Peng, Hadar Averbuch-Elor, Xiaowei Zhou, and Noah Snavely. 2023. Neural Scene Chronology. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 20752–20761.
- Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. 2020. Neural Sparse Voxel Fields. In *NeurIPS*. <https://proceedings.neurips.cc/paper/2020/file/b4b758962f17808746e9bb832a6fa4b8-Paper.pdf>
- Lingjie Liu, Weipeng Xu, Michael Zollhoefer, Hyeyoung Kim, Florian Bernard, Marc Habermann, Wenping Wang, and Christian Theobalt. 2019b. Neural rendering and reenactment of human actor videos. *ACM TOG* (2019).
- Shichen Liu, Shunsuke Saito, Weikai Chen, and Hao Li. 2019a. Learning to infer implicit surfaces without 3d supervision. *NeurIPS* (2019).
- Yuan Liu, Sida Peng, Lingjie Liu, Qianqian Wang, Peng Wang, Christian Theobalt, Xiaowei Zhou, and Wenping Wang. 2021. Neural Rays for Occlusion-aware Image-based Rendering. *arXiv* (2021).
- Stephen Lombardi, Tomás Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. 2019. Neural volumes: Learning dynamic renderable volumes from images. In *SIGGRAPH*.
- Stephen Lombardi, Tomás Simon, Gabriel Schwartz, Michael Zollhoefer, Yaser Sheikh, and Jason Saragih. 2021. Mixture of volumetric primitives for efficient neural rendering. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–13.
- 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 TOG* (2019).
- 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. *SIGGRAPH* (2022).
- Thomas Neff, Pascal Stadelbauer, Mathias Panger, Andreas Kurz, Joerg H Mueller, Chakravarthy R Alla Chaitanya, Anton Kaplanyan, and Markus Steinberger. 2021. DONeRF: Towards Real-Time Rendering of Compact Neural Radiance Fields using Depth Oracle Networks. In *EGSR*.
- Richard A Newcombe, Dieter Fox, and Steven M Seitz. 2015. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *CVPR*.
- Sergio Orts-Escalano, Christoph Rhemann, Sean Fanello, Wayne Chang, Adarsh Kowdle, Yury Degtyarev, David Kim, Philip L Davidson, Sameh Khamis, Mingsong Dou, et al. 2016. Holoporation: Virtual 3d teleportation in real-time. In *UIST*.

- Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. 2021a. Nerfies: Deformable Neural Radiance Fields. In *ICCV*.
- Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M Seitz. 2021b. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *arXiv preprint arXiv:2106.13228* (2021).
- Sida Peng, Yunzhi Yan, Qing Shuai, Hujun Bao, and Xiaowei Zhou. 2023. Representing Volumetric Videos as Dynamic MLP Maps. In *CVPR*.
- Sida Peng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. 2021. Neural Body: Implicit Neural Representations with Structured Latent Codes for Novel View Synthesis of Dynamic Humans. In *CVPR*.
- Eric Penner and Li Zhang. 2017. Soft 3D reconstruction for view synthesis. *ACM TOG* (2017).
- Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. 2021. D-NeRF: Neural Radiance Fields for Dynamic Scenes. In *CVPR*.
- Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*.
- Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. 2021. KiloNeRF: Speeding Up Neural Radiance Fields With Thousands of Tiny MLPs. In *ICCV*. 14335–14345.
- Gernot Riegler and Vladlen Koltun. 2020. Free View Synthesis. In *ECCV*.
- Gernot Riegler and Vladlen Koltun. 2021. Stable view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12216–12225.
- Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*.
- Ruizhi Shao, Zerong Zheng, Han Zhang Tu, Boning Liu, Hongwen Zhang, and Yebin Liu. 2022. Tensor4D: Efficient Neural 4D Decomposition for High-fidelity Dynamic Reconstruction and Rendering. *arXiv* (2022).
- Meng-Li Shih, Shih-Yang Su, Johannes Kopf, and Jia-Bin Huang. 2020. 3d photography using context-aware layered depth inpainting. In *CVPR*.
- Vincent Sitzmann, Semon Reznikov, Bill Freeman, Josh Tenenbaum, and Fredo Durand. 2021. Light field networks: Neural scene representations with single-evaluation rendering. *Advances in Neural Information Processing Systems* 34 (2021), 19313–19325.
- Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhöfer. 2019. DeepVoxels: Learning Persistent 3D Feature Embeddings. In *CVPR*. <https://doi.org/10.1109/CVPR.2019.00254>
- Liangchen Song, Anpei Chen, Zhong Li, Zhang Chen, Lele Chen, Junsong Yuan, Yi Xu, and Andreas Geiger. 2023. Nerfplayer: A streamable dynamic scene representation with decomposed neural radiance fields. *IEEE Transactions on Visualization and Computer Graphics* 29, 5 (2023), 2732–2742.
- Pratul P Srinivasan, Richard Tucker, Jonathan T Barron, Ravi Ramamoorthi, Ren Ng, and Noah Snavely. 2019. Pushing the boundaries of view extrapolation with multiplane images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 175–184.
- Mohammed Suhail, Carlos Esteves, Leonid Sigal, and Ameesh Makadia. 2022. Light Field Neural Rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 8269–8279.
- Gary J Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand. 2012. Overview of the high efficiency video coding (HEVC) standard. *IEEE Transactions on circuits and systems for video technology* 22, 12 (2012), 1649–1668.
- Cheng Sun, Min Sun, and Hwann-Tzong Chen. 2021. Direct Voxel Grid Optimization: Super-fast Convergence for Radiance Fields Reconstruction. *arXiv preprint arXiv:2111.11215* (2021).
- Tiancheng Sun, Zexiang Xu, Xiuming Zhang, Sean Fanello, Christoph Rhemann, Paul Debevec, Yun-Ta Tsai, Jonathan T Barron, and Ravi Ramamoorthi. 2020. Light stage super-resolution: continuous high-frequency relighting. *ACM Transactions on Graphics (TOG)* 39, 6 (2020), 1–12.
- Richard Szeliski and Polina Golland. 1998. Stereo matching with transparency and matting. In *Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271)*. IEEE, 517–524.
- Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P Srinivasan, Jonathan T Barron, and Henrik Kretzschmar. 2022. Block-nerf: Scalable large scene neural view synthesis. In *CVPR*.
- Richard Tucker and Noah Snavely. 2020. Single-view view synthesis with multiplane images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 551–560.
- Haithem Turki, Deva Ramanan, and Mahadev Satyanarayanan. 2022. Mega-NeRF: Scalable Construction of Large-Scale NeRFs for Virtual Fly-Throughs. In *CVPR*.
- Ziyu Wan, Christian Richardt, Aljaž Božič, Chao Li, Vijay Rengarajan, Seonghyeon Nam, Xiaoyu Xiang, Tuotuo Li, Bo Zhu, Rakesh Ranjan, et al. 2023. Learning Neural Duplex Radiance Fields for Real-Time View Synthesis. *arXiv preprint arXiv:2304.10537* (2023).
- Liao Wang, Jiakai Zhang, Xinhong Liu, Fuqiang Zhao, Yanshun Zhang, Yingliang Zhang, Minye Wu, Lan Xu, and Jingyi Yu. 2022. Fourier PlenOctrees for Dynamic Radiance Field Rendering in Real-time. *CVPR* (2022).
- Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul Srinivasan, Howard Zhou, Jonathan T. Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. 2021. IBRNet: Learning Multi-View Image-Based Rendering. In *CVPR*.
- Shengze Wang, Alexey Supikov, Joshua Ratcliff, Henry Fuchs, and Ronald Azuma. 2023. INV: Towards Streaming Incremental Neural Videos. *arXiv preprint arXiv:2302.01532* (2023).
- Suttisak Wizadwongsu, Pakkapon Phongthawee, Jiraphon Yenphraphai, and Supasorn Suwajanakorn. 2021. Nex: Real-time view synthesis with neural basis expansion. In *CVPR*.
- Minye Wu, Yuehao Wang, Qiang Hu, and Jingyi Yu. 2020. Multi-View Neural Human Rendering. In *CVPR*.
- Wenqi Xian, Jia-Bin Huang, Johannes Kopf, and Changil Kim. 2021. Space-time neural irradiance fields for free-viewpoint video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9421–9431.
- Meng You and Junhui Hou. 2023. Decoupling Dynamic Monocular Videos for Dynamic View Synthesis. *arXiv preprint arXiv:2304.01716* (2023).
- Alex Yu, Saro Fridovich-Keil, Matthew Tancik, Qinzhong Chen, Benjamin Recht, and Angjoo Kanazawa. 2022. Plenoxels: Radiance Fields without Neural Networks. *CVPR* (2022).
- Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. 2021a. PlenOctrees for Real-Time Rendering of Neural Radiance Fields. In *ICCV*.
- Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. 2021b. pixelNeRF: Neural Radiance Fields from One or Few Images. In *CVPR*.
- Tao Yu, Zerong Zheng, Kaiwen Guo, Jianhui Zhao, Qionghai Dai, Hao Li, Gerard Pons-Moll, and Yebin Liu. 2018. Doublefusion: Real-time capture of human performances with inner body shapes from a single depth sensor. In *CVPR*.
- Jiakai Zhang, Xinhong Liu, Xinyi Ye, Fuqiang Zhao, Yanshun Zhang, Minye Wu, Yingliang Zhang, Lan Xu, and Jingyi Yu. 2021. Editable free-viewpoint video using a layered neural representation. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–18.
- Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. 2018. Stereo magnification: Learning view synthesis using multiplane images. In *SIGGRAPH*.
- C Lawrence Zitnick, Sing Bing Kang, Matthew Uyttendaele, Simon Winder, and Richard Szeliski. 2004. High-quality video view interpolation using a layered representation. *ACM TOG* (2004).

SUPPLEMENTARY MATERIAL

In the supplementary material, we provide implementation details, more visualization results and per-scene breakdown. Please refer to our video for more comparison results and visualization results.

A IMPLEMENTATION DETAILS

A.1 Loss Formulation

We optimize our scene representation using the mean squared error (MSE) loss with color images.

$$\mathcal{L}_{\text{mse}} = \frac{1}{N_p} \sum_{i=1}^{N_p} (C_i - \hat{C}_i)^2, \quad (5)$$

where N_p represents the number of sampled pixels for each training iteration, and C_i and \hat{C}_i are the ground truth and predicted colors of the i -th pixel, respectively. In addition to the color loss, we follow [Fridovich-Keil et al. 2023; Yu et al. 2022] and use the total variation (TV) loss to regularize the explicit feature planes $\{\mathbf{P}_i | i \in \{xy, xz, yz, xt, yt, zt\}\}$. We detail our loss formulation in the supplementary material. The TV loss can be formulated as:

$$\mathcal{L}_{\text{tv}}(\mathbf{P}) = \sum_{j=1}^{h-1} \sum_{k=1}^{w-1} \left((\mathbf{P}(j+1, k) - \mathbf{P}(j, k))^2 + (\mathbf{P}(j, k+1) - \mathbf{P}(j, k))^2 \right), \quad (6)$$

where h and w denote the height and width of the explicit feature plane \mathbf{P} , respectively. We sum the TV losses for the six feature planes as the final TV loss. The TV loss is assigned a weight of 0.001, while the MSE loss has a weight of 1 in the final loss calculation.

A.2 Evaluation Details

We include the evaluation details for the DyNeRF dataset [Li et al. 2021b] in Fig. 6. For a 10-second clip, we evaluate 10 frames and Fig. 6 presents one of the frames.

A.3 Other Details

Our feature planes have multiple spatial (xy, yz, xz) resolutions ($512 \times 512, 256 \times 256, 128 \times 128, 64 \times 64$) and one temporal (xt, yt, zt) resolution ($\text{spatial_resolution} \times \frac{\text{frame_number}}{2}$). The number of channels for each feature plane is 16. These feature planes We use concatenation to aggregate multi-resolution features. The CNN feature network's architecture is the same as the architecture used in ENeRF [Lin et al. 2022]. The hidden size of m_c and m_σ is 64, and the number of hidden layers is 1. We use ReLU as the activation function. Our appearance-related networks can also be pre-trained with generalizable radiance field methods [Lin et al. 2022; Wang et al. 2021]. Specifically, we slightly modify the appearance rendering head of ENeRF [Lin et al. 2022] to align it with our method and retrain it on the DTU dataset. To train our method on a new scene, we will load this pre-trained appearance network. We experimentally found that this helps our method get slight better results as shown in the ablations in the main paper. We optimize our model using Adam [Kingma and Ba 2015] optimizer with an initial learning rate of 0.01 for explicit plane features and 0.001 for MLPs and the feature network. The learning rate is decayed with a cosine learning rate decay strategy and will be decayed into zero when training is finished. (100k iterations on the ZJU-MoCap and



Figure 6: Evaluation details on the DyNeRF dataset. The left image is the first frame (test frame) of a one-second video clip, and the right image is the average frame of this second. We identify the 6 patches with the largest differences between the test frame and the average frame. During the quantitative evaluation, we only assess these patches.

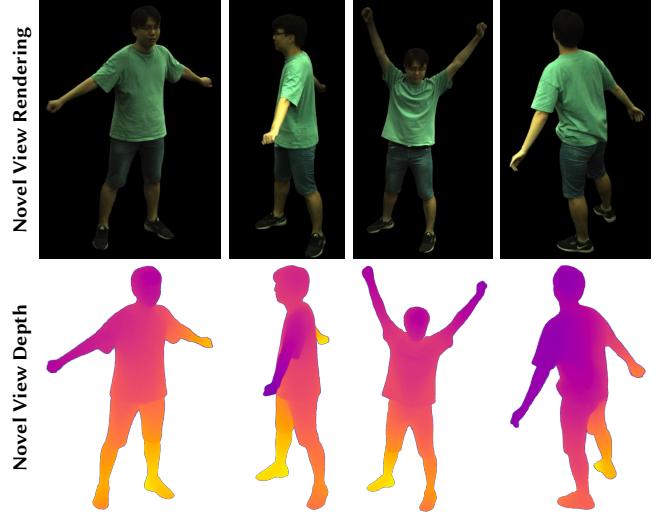


Figure 7: Novel view synthesis results on the ZJU-MoCap dataset. These images are taken from a video generated by rendering a trajectory from novel perspectives. Please watch the complete video sequence in the supplementary video. The high-quality depth results indicate that our method can achieve inter-view rendering consistency.

NHR datasets, 140k on the DNA-Rendering datasets, 15K on the DyNeRF dataset.) For each training iteration, we randomly select 4 images and sample 1024 pixels from each image as a training batch. Following [Barron et al. 2021], we use an additional proposal m_σ with feature planes of a single spatial resolution (128) to propose points. For each ray, we uniformly sample 64 points to infer the proposal network to get 32 proposal points. We implement our efficient rendering strategy using NerfAcc [Li et al. 2023].

B ADDITIONAL RESULTS

We include qualitative comparison results on the ZJU-MoCap and NHR datasets in Fig. 8. We provide additional results on the ZJU-MoCap dataset in Figure 7. The high-quality depth results we

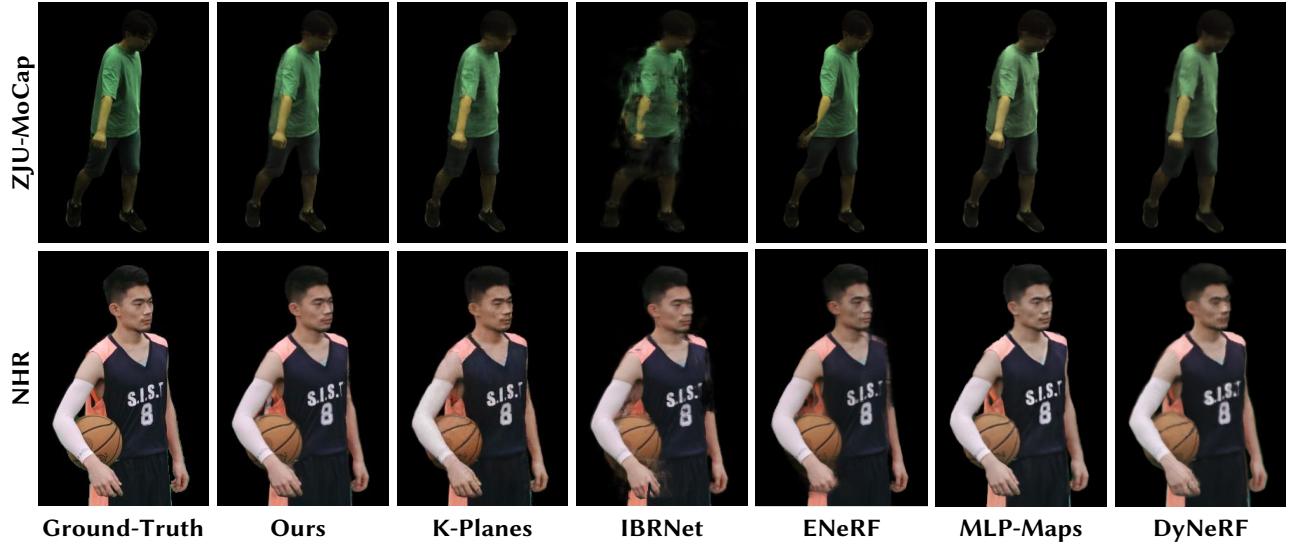


Figure 8: Qualitative comparison of image synthesis results on the ZJU-MoCap and NHR datasets.

achieved are the reason why our method is able to provide consistent rendering.

C PER-SCENE BREAKDOWN

Tables 5, 6 present the per-scene comparisons. These results are consistent with the averaged results in the paper.

Table 5: Quantitative comparison of view synthesis results on the DNA-Rendering dataset.

	0008_03	0013_01	0013_03	0013_09	0008_03	0013_01	0013_03	0013_09	0008_03	0013_01	0013_03	0013_09
	PSNR↑				SSIM↑				LPIPS↓			
K-Planes _{2h}	26.94	26.00	25.91	26.53	0.930	0.943	0.944	0.957	0.185	0.118	0.114	0.117
K-Planes _{8h}	27.99	26.80	27.87	27.15	0.940	0.950	0.958	0.960	0.165	0.107	0.091	0.108
IBRNet	25.76	26.08	25.15	24.28	0.945	0.960	0.952	0.959	0.150	0.087	0.091	0.097
IBRNet _{3.5h}	27.66	28.84	27.53	27.35	0.955	0.973	0.965	0.973	0.127	0.060	0.067	0.070
ENeRF	26.79	27.92	26.84	25.88	0.955	0.973	0.965	0.969	0.109	0.055	0.061	0.069
ENeRF _{3.5h}	27.74	28.99	28.08	27.84	0.958	0.976	0.970	0.970	0.113	0.048	0.052	0.052
Ours _{0.51h}	27.70	27.62	26.46	26.80	0.952	0.966	0.960	0.968	0.121	0.067	0.069	0.077
Ours _{3.33h}	28.90	29.53	28.75	28.78	0.962	0.977	0.972	0.979	0.096	0.047	0.052	0.051

Table 6: Quantitative comparison of view synthesis results on ZJU-MoCap and NHR datasets.

	ZJU-MoCap									NHR			
	313	315	377	386	387	390	392	393	394	basketball	sport_1	sport_2	sport_3
	PSNR↑									PSNR↑			
DyNeRF _{>24h}	31.50	30.29	28.92	30.88	27.90	30.14	30.09	29.28	29.88	27.97	31.76	32.43	31.33
MLP-Maps _{>24h}	32.15	29.94	29.40	31.05	27.89	30.10	31.06	29.78	30.15	29.11	32.92	33.19	33.59
K-Planes _{2h}	30.85	29.23	28.69	30.67	27.43	29.84	30.12	28.85	29.82	28.26	32.23	30.70	30.89
K-Planes _{8h}	32.11	30.55	29.40	30.82	27.75	30.06	31.00	29.61	30.17	30.01	34.52	33.96	33.24
IBRNet	29.08	25.13	27.47	29.97	26.27	28.59	28.90	27.86	28.18	27.01	28.91	29.94	28.68
IBRNet _{2.5h}	30.89	28.12	27.47	30.66	27.43	29.84	30.12	28.85	29.82	30.47	34.65	35.02	34.01
ENeRF	30.31	28.13	28.73	30.34	27.24	29.32	29.86	28.84	29.18	25.98	25.87	27.40	26.30
ENeRF _{2.5h}	30.69	28.79	28.51	30.10	27.32	29.09	30.03	29.07	29.29	28.22	30.68	32.08	31.26
Ours _{0.49h}	31.62	30.09	29.24	30.89	27.89	30.16	30.84	29.59	30.27	29.51	33.67	34.21	32.21
Ours _{2.29h}	32.87	30.75	29.61	30.82	28.16	30.31	31.38	30.00	30.52	30.57	34.87	35.61	33.82
	SSIM↑									SSIM↑			
DyNeRF _{>24h}	0.970	0.976	0.960	0.960	0.953	0.959	0.953	0.952	0.952	0.929	0.954	0.945	0.944
MLP-Maps _{>24h}	0.976	0.977	0.963	0.960	0.953	0.959	0.962	0.958	0.957	0.943	0.959	0.954	0.956
K-Planes _{2h}	0.968	0.975	0.960	0.955	0.947	0.955	0.954	0.948	0.950	0.933	0.958	0.944	0.940
K-Planes _{8h}	0.975	0.981	0.965	0.957	0.952	0.957	0.960	0.955	0.955	0.949	0.968	0.960	0.956
IBRNet	0.941	0.930	0.939	0.942	0.931	0.936	0.936	0.931	0.926	0.926	0.945	0.933	0.935
IBRNet _{2.5h}	0.968	0.968	0.960	0.954	0.948	0.952	0.958	0.951	0.946	0.961	0.969	0.964	0.965
ENeRF	0.968	0.969	0.964	0.958	0.955	0.956	0.959	0.956	0.954	0.927	0.943	0.923	0.930
ENeRF _{2.5h}	0.970	0.971	0.960	0.954	0.953	0.953	0.959	0.956	0.955	0.938	0.956	0.947	0.949
Ours _{0.49h}	0.977	0.981	0.965	0.960	0.956	0.960	0.962	0.957	0.958	0.958	0.969	0.963	0.960
Ours _{2.29h}	0.982	0.983	0.968	0.959	0.958	0.961	0.965	0.960	0.960	0.966	0.976	0.971	0.968
	LPIPS↓									LPIPS↓			
DyNeRF _{>24h}	0.070	0.061	0.083	0.082	0.094	0.083	0.113	0.102	0.099	0.142	0.095	0.119	0.114
MLP-Maps _{>24h}	0.049	0.047	0.069	0.072	0.081	0.068	0.074	0.080	0.072	0.094	0.067	0.084	0.076
K-Planes _{2h}	0.085	0.070	0.106	0.125	0.117	0.100	0.124	0.118	0.117	0.164	0.103	0.140	0.135
K-Planes _{8h}	0.056	0.045	0.081	0.108	0.094	0.086	0.090	0.090	0.091	0.133	0.076	0.097	0.099
IBRNet	0.116	0.124	0.121	0.132	0.126	0.129	0.132	0.126	0.129	0.123	0.094	0.123	0.114
IBRNet _{2.5h}	0.066	0.070	0.078	0.094	0.093	0.095	0.089	0.090	0.083	0.079	0.068	0.087	0.077
ENeRF	0.042	0.043	0.046	0.054	0.053	0.056	0.056	0.056	0.053	0.095	0.072	0.098	0.088
ENeRF _{2.5h}	0.036	0.039	0.046	0.052	0.053	0.056	0.051	0.055	0.049	0.088	0.061	0.077	0.075
Ours _{0.49h}	0.040	0.034	0.063	0.068	0.068	0.064	0.071	0.073	0.068	0.081	0.059	0.079	0.075
Ours _{2.29h}	0.027	0.027	0.046	0.063	0.055	0.056	0.055	0.058	0.055	0.061	0.043	0.060	0.058