Learning Compositional Radiance Fields of Dynamic Human Heads

Ziyan Wang[1,3] Timur Bagautdinov[3] Stephen Lombardi[3] Tomas Simon[3]
Jason Saragih[3] Jessica Hodgins[1,2] Michael Zollhöfer[3]
[1]Carnegie Mellon University [2]Facebook AI Research [3]Facebook Reality Labs

https://ziyanw1.github.io/hybrid_nerf/

Abstract

Photorealistic rendering of dynamic humans is an important ability for telepresence systems, virtual shopping, synthetic data generation, and more. Recently, neural rendering methods, which combine techniques from computer graphics and machine learning, have created high-fidelity models of humans and objects. Some of these methods do not produce results with high-enough fidelity for driveable human models (Neural Volumes) whereas others have extremely long rendering times (NeRF). We propose a novel compositional 3D representation that combines the best of previous methods to produce both higher-resolution and faster results. Our representation bridges the gap between discrete and continuous volumetric representations by combining a coarse 3D-structure-aware grid of animation codes with a continuous learned scene function that maps every position and its corresponding local animation code to its view-dependent emitted radiance and local volume density. Differentiable volume rendering is employed to compute photo-realistic novel views of the human head and upper body as well as to train our novel representation end-to-end using only 2D supervision. In addition, we show that the learned dynamic radiance field can be used to synthesize novel unseen expressions based on a global animation code. Our approach achieves state-of-the-art results for synthesizing novel views of dynamic human heads and the upper body.

1. Introduction

Modeling, rendering, and animating dynamic human heads at high fidelity, for example for virtual reality remote communication applications, is a highly challenging research problem. The main reason for this is the tremendous complexity of the human head in terms of geometry and appearance variations, e.g., of human skin, hair, teeth, and the eyes. Skin exhibits subsurface scattering and shows fine-scale geometric pore-level detail, while the human eyes and teeth are both translucent and reflective at the same time. High fidelity modeling and rendering of human hair is challenging due to its thin geometric structure and light scattering properties. Importantly, the face is not static, but changes dynamically with expression and posture.

Recent work on neural rendering proposes to learn either discrete or continuous neural scene representations to achieve viewpoint and animation controllable rendering. Discrete neural scene representations are based on meshes [32, 15, 19, 8, 31], point clouds [34, 1, 21], voxel grids [17, 29], or multi-plane images [36, 22]. However, each of these representations has drawbacks: Meshes, even if dynamically textured [16], struggle to model thin and detailed structures, such as hair. Point clouds, by design, do not provide connectivity information and thus lead to undefined signals in areas of sparse sampling, while making explicit occlusion reasoning challenging. Multi-plane images yield photo-realistic rendering results under constrained camera motion, but produce 'stack of cards'-like artifacts [36] when the camera moves freely. Volumetric representations [17] based on discrete uniform voxel grids are capable of modeling thin structures, e.g., hair, using semi-transparency. While these approaches achieve impressive results, they are hard to scale up due to their innate cubic memory complexity.

To circumvent the cubic memory complexity of these approaches, researchers have proposed continuous volumetric scene representations based on fully-connected networks that map world coordinates to a local feature representation. Scene Representation Networks (SRNs) [30] employ sphere marching to extract the local feature vector for every point on the surface, before mapping to pixel colors. While this paper showed very inspiring results, the approach is limited to modeling diffuse objects, which makes it unsuitable to represent human heads at high fidelity.

Neural radiance fields [23] have shown impressive results for synthesizing novel views of static scenes at impressive accuracy by mapping world coordinates to view-dependent emitted radiance and local volume density. A very recent extension [14] speeds up rendering by applying

a static Octree to cull free space. While they have shown first results on a simple synthetic dynamic sequence, it is unclear how to extend the approach to learn and render photo-realistic dynamic sequences of real humans. In addition, it is unclear how to handle expression interpolation and the synthesis of novel unseen motions given the static nature of the Octree acceleration structure.

As discussed, existing work on continuous neural 3D scene representations mainly focuses on static scenes, which makes dynamic scene modeling and editing not directly achievable under the current frameworks. In this work, we propose a novel compositional 3D scene representation for learning high-quality dynamic neural radiance fields that addresses these challenges. To this end, we bridge the gap between discrete and continuous volumetric representations by combining a coarse 3D-structure-aware grid of animation codes with a continuous learned scene function. We start by extracting a global animation code from a set of input images using a convolutional encoder network. The global code is then mapped to a 3D-structure-aware grid of local animation codes as well as a coarse opacity field. A novel importance sampling approach employs the regressed coarse opacity to speed up rendering. To facilitate generalization across motion and shape/appearance variation, in addition to conditioning the dynamic radiance field on the global animation code, we additionally condition it on a local code which is sampled from the 3D-structure-aware grid of animation codes. The final pixel color is computed by volume rendering. In summary, the main contributions of our work are:

- A novel compositional 3D representation for learning high-quality dynamic neural radiance fields of human heads in motion based on a 3D-structure-aware grid of local animation codes.
- An importance sampling strategy tailored to human heads that allows to remove unnecessary computation in free space and enables faster volumetric rendering.
- State-of-the-art results for synthesizing novel views of dynamic human heads that outperform competing methods in terms of quality.

2. Related Work

Recently, there have been many works that combine deep neural networks with geometric representations to perform rendering. In this section, we discuss different methods and their trade-offs, categorized by their underlying geometric representation.

Mesh-based Representations: Triangle meshes have been used for decades in computer graphics since they provide an explicit representation of a 2D surface embedded within a 3D space. A primary benefit of this representation is the ability to use high-resolution 2D texture maps to model high-frequency detail on flat surfaces. Recently, differentiable rasterization [12, 15, 6, 8, 18, 19, 31] has made it possible to jointly optimize mesh vertices and texture using gradient descent based on a 2D photometric re-rendering loss. Unfortunately, these methods often require a good initialization of the mesh vertices or strong regularization on the 3D shape to enable convergence. Moreover, these methods require a template mesh with fixed topology which is difficult to acquire.

Point Cloud-based Representations: Point clouds are an explicit geometric representation that lacks connectivity between points, alleviating the requirement of a fixed topology but losing the benefits of 2D texture maps for appearance modeling. Recent works, like [21] and [1], propose methods that generate photo-realistic renderings using an image-to-image translation pipeline that takes as input a deferred shading deep buffer consisting of depth, color, and semantic labels. Similarly, in SynSin [34], per-pixel features from a source image are lifted to 3D to form a point cloud which is later projected to a target view to perform novel view synthesis. Although point clouds are a lightweight and flexible geometric scene representation, rendering novel views using point clouds results in holes due to their inherent sparseness, and it typically requires image-based rendering techniques for in-painting and refinement.

Multi-plane Image-based Representations: Another line of work is using multi-plane images (MPIs) as the scene representation. MPIs [36] are a method to store color and alpha information at a discrete set of depth planes for novel view synthesis, but they only support a restricted range of motion. LLFF [22] seeks to enlarge the range of camera motion by fusing a collection of MPIs [36]. Multi-sphere images (MSIs) [2, 3] are an extension for the use case of stereo 360° imagery in VR, where the camera is located close to the center of a set of concentric spheres.

Voxel-based Representations: One big advantage of voxel-based representations is that they do not require precomputation of scene geometry and that they are easy to optimize with gradient-based optimization techniques. Many recent works [11, 33, 7, 35] have proposed to learn volumetric scene representation based on dense uniform grids. Recently, such volumetric representations have attracted a lot of attention for novel view synthesis. DeepVoxels [29] learns a persistent 3D feature volume for view synthesis with an image-based neural renderer. Neural Volumes [17] proposes a differentiable raymarching algorithm for optimizing a volume, where each voxel contains an RGB and transparency values. The main challenge for voxel-based techniques originates in the cubic memory complexity of the often employed dense uniform voxel grid, which makes

it hard to scale these approaches to higher resolutions.

Implicit Geometry Representations: Implicit geometry representations have drawn a lot of attention from the research community due to their low storage requirements and the ability to provide high-quality reconstructions with good generalization. This trend started with geometric reconstruction approaches that first employed learned functions to represent signed distance fields (SDFs) [25, 10, 4] or occupancy fields [20, 9, 26]. DeepSDF [25] and OccNet [20] are among the earliest works that try to learn an implicit function of a scene with an MLP and are fueled by large scale 3D shape datasets, such as ShapeNet [5]. DeepSDF densely samples points around the surface to create direct supervision for learning the continuous SDF, while OccNet learns a continuous occupancy field. ConvOccNet [26] manages to improve OccNet's ability to fit large scale scenes by introducing a convolutional encoder-decoder. ConvOccNet is limited to static scenes and geometry modeling, i.e., it can not handle the dynamic photo-realistic sequences that are addressed by our approach.

Continuous Scene Representations: Inspired by their implicit geometry counterparts, continuous scene representations for modeling colored scenes have been proposed. Scene Representation Networks (SRNs) [30] propose an approach to model colored objects by training a continuous feature function against a set of multi-view images. DVR [24] derived an analytical solution for the depth gradient to learn an occupancy and texture field from RGB images with implicit differentiation. NeRF [23] learns a 5D neural radiance field using differentiable raymarching by computing an integral along each ray. Although promising, their results are limited to a single static scene and the approach is hard to generalize to multiple scenes or a scene with dynamic objects. Another limiting factor is that these representations are extremely costly to render, since every step along the ray requires an expensive evaluation of the complete fully-connected network. GRAF [28] introduces a generative model for radiance fields which extends NeRF's ability to model multiple static objects. However, their approach is limited to very simple scenes at low resolution. The approach has not been demonstrated to scale to the dynamic high-quality real-world animations we are interested in. A very recent work, NSVF [14], manages to solve the second limitation with an Octree acceleration structure. Although they provide initial results on a single synthetic dynamic sequence, the static Octree structure is optimized frame by frame rather than regressed from temporal information, which makes it not straightforward to directly deploy their method on novel photo-realistic dynamic sequences of real humans. In addition, it is unclear how to efficiently handle expression interpolation and novel motion synthesis given their static Octree.

3. Method

In this section, we introduce our novel compositional representation that combines the modeling power of high-capacity voxel-based representations and the ability of continuous scene representations to capture subtle fine-level detail. In Fig. 1 we provide an overview of our approach.

The core of the method is a hybrid encoder-decoder architecture, directly supervised with multi-view video sequences. For a given frame, the encoder takes a sparse set of views, and outputs a global animation code, which describes dynamic scene information specific to the frame. The global animation code is used to condition the 3D convolutional decoder, which outputs a coarse 3D structure-aware voxel field. In particular, each voxel stores coarse-level opacity, color and localized animation codes, which represent local dynamical properties of the corresponding spatial region of the scene. The resulting voxel field is further used to create a coarse volumetric rendering of the scene, which may lack fine-level detail, but provides a reliable initial estimate of the scene's geometry, which is crucial to enable efficient continuous scene modeling. To account for the lack of detail, we rely on a continuous scene function, represented as an MLP, to model fine-level radiance. The coarse-level geometry estimate is used to define spatial regions where the function is evaluated, and the local animation codes as spatially-varying conditioning signal to the MLP. To better model view-dependent effects, both coarse- and fine-level representations are partly conditioned on the camera viewpoint. The outputs of the continuous scene function are then used to create the final, refined volumetric rendering of the scene. In what follows, we describe each of the components in detail.

3.1. Encoder-Decoder

The goal of the encoder is to produce a compact representation that captures global dynamical properties of the scene, which then serves as a conditioning signal for the decoder. Our encoder is a 2D convolutional network which takes a sparse set of views and outputs parameters of a diagonal Gaussian distribution $\boldsymbol{\mu}, \boldsymbol{\sigma} \in \mathbb{R}^{256}$. In practice, the encoder is conditioned on three different camera views, concatenated along the channel axis. Given the distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma})$, we use the reparameterization trick to produce the global animation code $\mathbf{z} \in \mathbb{R}^{256}$ in a differentiable way, and pass it to the decoder. We found that using a variational formulation [13] is critical for making our model animatable.

Given the global animation code $\mathbf{z}$, the goal of the decoder is to produce a coarse-level representation of the scene. In particular, the coarse level is modeled by a volumetric field

$$\mathbf{V}_{\mathbf{p}} = \left(\widetilde{\mathbf{c}}_{\mathbf{p}}, \widetilde{\sigma}_{\mathbf{p}}, \mathbf{f}_{\mathbf{p}}, \mathbf{f}_{\mathbf{p}}^{v}\right) \ , \tag{1}$$

where $\widetilde{\mathbf{c}}_{\mathbf{p}} \in \mathbb{R}^3$ is a coarse-level color value, $\widetilde{\sigma}_{\mathbf{p}} \in \mathbb{R}$ is

enc输入稀疏视图，生成一个256维高斯分布，然后根据高斯分布生成全局动画编码z，输入dec得到动画

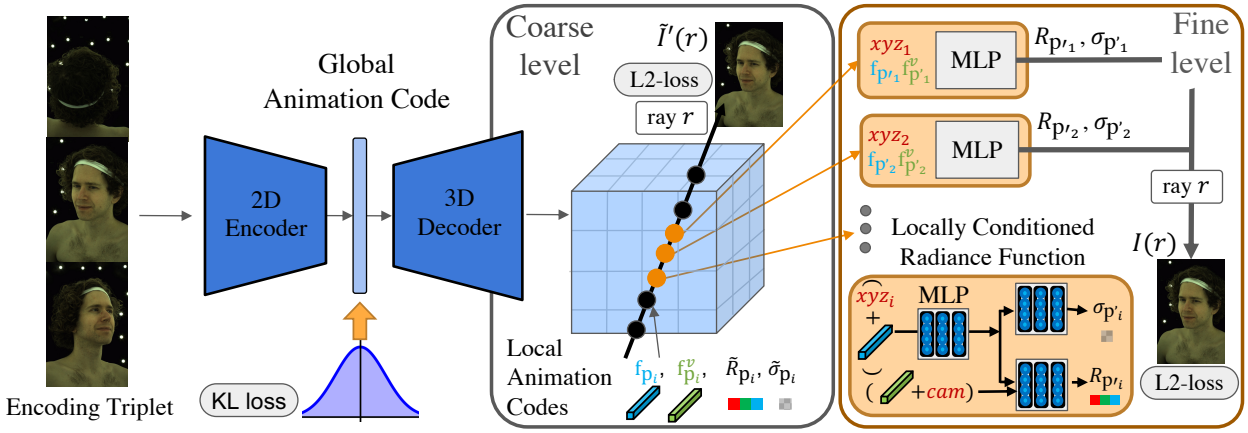

Figure 1. **Method overview**. Given a multi-view video as input, we learn a dynamic radiance field parametrized by a global animation code. To render a particular frame, the global code is first mapped to a coarse voxelized field of local animation codes using a 3D convolutional decoder. This grid of animation codes provides local conditioning at each 3D position for the fine-level radiance function, represented as an MLP. Differentiable ray marching is used to render images and provide supervision, and can be sped up significantly by using a ray sampling strategy that uses the coarse grid to determine relevant spatial regions.

differential opacity, $\mathbf{f_p} \in \mathbb{R}^{32}$ is the view-independent local animation code, $\mathbf{f}^v_\mathbf{p} \in \mathbb{R}^{32}$ is the view-dependent local animation code, and $\mathbf{p} \in \mathbb{R}^3$ is the spatial location. In our framework, $\mathbf{V}$ is produced by a volumetric decoder as an explicit coarse discrete grid $\mathbf{G} \in \mathbb{R}^{D\times D\times D\times F}$, where $D = 64$ is the spatial dimension of the grid, and $F = 68$ is the dimensionality of the field. Samples $\mathbf{V_p}$ at continuous locations $\mathbf{p} \in \mathbb{R}^3$ are produced with trilinear interpolation over the voxels.

In practice, the decoder is represented by two independent 3D convolutional neural network branches. The first branch is conditioned only on the global code $\mathbf{z}$, and predicts view-independent values, the differential occupancy $\widetilde{\sigma}_\mathbf{p}$ and the view-independent local animation codes $\mathbf{f_p}$. The second branch predicts view-dependent color values $\widetilde{\mathbf{c}}_\mathbf{p}$ and local animation codes $\mathbf{f}^v_\mathbf{p}$, and is conditioned on both the global code $\mathbf{z}$ and the viewpoint $\mathbf{v} \in \mathbb{R}^3$, which is computed as a normalized difference between the camera location and the center of the scene.

3.2. Volumetric Rendering

Given the discrete voxel field, we apply differentiable ray-marching to obtain coarse volumetric rendering [23]. Namely, for each ray $\mathbf{r} \in \mathbb{R}^3$ shot from the camera center $\mathbf{o} \in \mathbb{R}^3$, we sample N query points $\mathbf{p}_i = (\mathbf{o} + d_i \cdot \mathbf{r})$ along $\mathbf{r}$, where d_i is the depth sampled uniformly between the depth at a near plane d_{min} and a far plane d_{max}. Estimates of expected coarse opacity $\widetilde{A}_\mathbf{r}$ and color $\widetilde{I}'_\mathbf{r}$ are then computed as

$$\widetilde{A}_\mathbf{r} = \sum_{i=1}^{N} T_i \alpha_i \,, \qquad \widetilde{I}'_\mathbf{r} = \sum_{i=1}^{N} T_i \alpha_i \widetilde{\mathbf{c}}_{\mathbf{p}_i} \,, \tag{2}$$

where $T_i = \exp(-\sum_{j=1}^{i-1} \widetilde{\sigma}_{\mathbf{p}_j}\delta_j)$, $\alpha_i = (1-\exp(-\widetilde{\sigma}_{\mathbf{p}_i}\delta_i))$, and $\delta_i = \|d_{i+1} - d_i\|$ is the distance between two neighbouring depth samples. In practice, values $\widetilde{\mathbf{c}}_{\mathbf{p}_i}, \widetilde{\sigma}_{\mathbf{p}_i}$ are sampled from the voxel grid with trilinear interpolation.

The final coarse-level rendering is computed by compositing the accumulated color $\widetilde{I}'_\mathbf{r}$ and the background color with a weighted sum

$$\widetilde{I}_\mathbf{r} = \widetilde{I}'_\mathbf{r} + (1 - \widetilde{A}_\mathbf{r}) I^{bg}_\mathbf{r} \,. \tag{3}$$

The resulting coarse rendering roughly captures the appearance of the scene, but lacks fine-level detail. A seemingly straightforward way to improve the level of detail would be to increase the spatial resolution of the voxel grid. Unfortunately, this quickly becomes impractical due to the cubic memory complexity of these representations.

3.3. Continuous Scene Function

In order to improve fine-level modeling capabilities while avoiding heavy memory costs associated with high-res voxel representations, we introduce a *continuous* scene function $f(\cdot)$, parameterized as an MLP. The key intuition is that voxel-based approaches represent scenes *explicitly* and uniformly across space, thus often wasting resources on irrelevant areas. On the other hand, continuous representations are *implicit*, and allow for more flexibility, as the scene function can be evaluated at arbitrary locations. When combined with a sampling strategy that focuses only on relevant spatial locations, this flexibility can bring significant efficiency improvements.

One crucial difference of our method with respect to the existing continuous neural rendering approaches [28, 23], is that in addition to conditioning on the location, view direction and the global scene information, our scene function is also conditioned on spatially-varying local animation

codes. As we demonstrate in our experiments in Sec. 4, this increases the effective capacity of our model, and allows our model to capture significantly more detail and better generalize across different motion and shape/appearance variations. We also show that this is especially important for modeling dynamic scenes, as they require significantly more modeling capacity and the naive MLP-based approaches typically fail.

More formally, the scene function $f(\cdot)$ takes as inputs coordinates of a sampled query point $\mathbf{p}$, view vector $\mathbf{v}$, and the corresponding local animation codes $\mathbf{f_p}, \mathbf{f_p^v}$, and produces the fine-level color $\mathbf{c_p} \in \mathbb{R}^3$ and the differential probability of opacity $\sigma_\mathbf{p} \in \mathbb{R}$

$$\mathbf{c_p}, \sigma_\mathbf{p} = f(\phi(\mathbf{p}), \phi(\mathbf{v}), \mathbf{f_p}, \mathbf{f_p^v}) \ .$$

Feature vectors $\mathbf{f_p}, \mathbf{f_p^v}$ are obtained from the the coarse voxel grid via trilinear interpolation, and position $\mathbf{p}$ and view $\mathbf{v}$ vectors are passed through a positional encoding $\phi(\cdot)$, in order to better capture high-frequency information [23].

Fine-level rendering $I_\mathbf{r}$ and $A_\mathbf{r}$ can then be computed by evaluating $f(\cdot)$ at a number of sampled query points along each ray and applying Eq. (2)-(3). In the next section, we discuss our novel sampling scheme that allows to significantly speed up the rendering process.

3.4. Efficient Sampling

Using spatially-varying conditioning allows us to increase effective capacity of our continuous scene representation and leads to better generalization. However, producing a high-quality rendering still requires evaluating the scene function at a large number of query locations, which can be computationally expensive [23], and ultimately suffers from similar limitations as the voxel fields. Luckily, we can exploit the fact that our coarse voxel field already contains information about the scene's geometry. To this end, we introduce a simple and efficient sampling scheme, which uses the coarse opacity values to produce a strong initial prior on the underlying geometry. In particular, for each ray $\mathbf{r}$, we first compute a coarse depth $\widetilde{d}_\mathbf{r}$ as

$$\widetilde{d}_\mathbf{r} = \frac{1}{\widetilde{A}_\mathbf{r}} \sum_{i=1}^{N} T_i \alpha_i \cdot d_i \ ,$$

where d_i are the *same* uniform samples as in Eq. (2). Then, we obtain our new fine-level location samples from a uniform distribution:

$$d \sim \mathcal{U}\left[\widetilde{d}_\mathbf{r} - \Delta_d, \widetilde{d}_\mathbf{r} + \Delta_d\right] \ ,$$

centered at the depth estimate $\widetilde{d}_\mathbf{r}$, where $\Delta_d = \frac{(d_{max} - d_{min})}{k}$, i.e. $k = 10$ times smaller range than at the coarse level. In Sec. 4 we demonstrate that this strategy in practice leads to comparable rendering quality, while being more computationally efficient.

3.5. Training Objective

Our model is end-to-end differentiable, which allows us to jointly train our encoder, decoder and the scene MLP, by minimizing the following loss:

$$\mathcal{L} = \mathcal{L}_r + \widetilde{\mathcal{L}}_r + \lambda_f \mathcal{L}_\beta + \lambda_c \widetilde{\mathcal{L}}_\beta + \lambda_{\text{KL}} \mathcal{L}_{\text{KL}} \ .$$

Here $\mathcal{L}_r$ is the error between the rendered and ground truth images for the fine-level rendering:

$$\mathcal{L}_r = \sum_{\mathbf{r} \in \mathcal{R}} ||I_\mathbf{r} - I_\mathbf{r}^{gt}||_2^2 \ ,$$

where $\mathcal{R}$ is a set of rays sampled in a batch. The coarse-level rendering loss $\widetilde{\mathcal{L}}_r$ is computed similarly. $\mathcal{L}_\beta$ and $\widetilde{\mathcal{L}}_\beta$ are the priors on the fine-level and coarse-level image opacities respectively [17]:

$$\mathcal{L}_\beta = \sum_{\mathbf{r} \in \mathcal{R}} \left(\log A_\mathbf{r} + \log(1 - A_\mathbf{r})\right) \ ,$$

which pushes both the coarse and fine opacities to be sharper, and encodes the prior belief that most of the rays should hit either the object or the background. Finally, the Kullback-Leibler divergence loss $\mathcal{L}_{\text{KL}}$ encourages our global latent space to be smooth [13], which improves the animation and interpolation capabilities of our model.

4. Experiments

We first compare with two state-of-the-art methods for novel view synthesis, namely NV [17] and NeRF [23] on four dynamic sequences of a human head making different facial expressions or talking. We then perform an ablation study to test how different feature representations affect the ability to capture longer sequences, as well as the effects of applying different resampling strategies on speed and image quality. We also evaluate generalization capabilities of our model on novel sequence generation and animation, by interpolating in latent space and by driving the model with various input modalities, including keypoints and images.

4.1. Datasets

We use a multi-camera system with around 100 synchronized color cameras that produces 2048×1334 resolution images at 30 Hz. The cameras are distributed approximately spherically at a distance of one meter, and focused at the center of the capture system to provide as many viewpoints as possible. Camera intrinsics and extrinsics are calibrated in an offline process. Images are downsampled to 1024×667 for training and testing. Each capture contains $n = 3$ sentences and around $k = 350$ frames in total for each camera view. We trained on $m = 93$ cameras and tested on $q = 33$ frames from another $p = 7$ cameras.

4.2. Baselines

We compare our methods with two baselines that we describe in the following.

NV [17]: Neural Volumes performs novel view synthesis of a dynamic object-centric scene by doing raymarching on a warped voxel grid of RGB and differential opacity that is regressed from three images using an encoder-decoder network. As the volume is conditioned on temporal input of RGB images, NV is capable of rendering dynamic scenes. The volume is of size 128^3 and the warp field is 32^3. The global animation code is a feature vector of 256 entries.

NeRF [23]: NeRF learns a continuous function of scene radiance, including RGB and opacity, with a fully connected neural network conditioned on scene coordinates and viewing direction. Positional encoding is applied to the 3D coordinates to better capture high frequency information, and raymarching is performed to render novel views. Note that the original NeRF approach is not directly applicable to dynamic sequences. Thus, we extend the conditioning signal to NeRF with a global animation code generated from the encoder in NV. The global animation code is generated from the encoder in NV and it is also of size 256.

4.3. Novel View Synthesis

We show quantitative and qualitative results of novel view synthesis on four dynamic sequences of human heads.
Quantitative Results: We report quantitative evaluation results in Tab. 1. Metrics used here are MSE, PSNR, and SSIM. We average those metrics across different test views and time steps, among each of the sequences. To compensate for sensory difference between each camera, we apply the same color calibration network as in NV [17] for our methods as well as all baselines. To compute the parameters of the color calibration networks, we first fit the color calibration model on an additional sentence with all camera views and fix the parameters for all subsequent steps. The first three sequences (Seq1-Seq3) are captures showing the participant talking, while the last one (Seq4) is a capture of a range of motions showing challenging expressions. As we can see, our method outperforms all other baselines on the four dynamic sequences in terms of all metrics.
Qualitative Results: We show visual comparisons between different models trained on long video sequences in Fig. 2. We can see that NV and NeRF trained on a sequence tend to yield relatively blurry results, while our approach produces sharper images and can reconstruct finer details in terms of both texture and geometry on areas like hair, eyes, and teeth. Our method can achieve photo-realistic rendering results on video sequences.

4.4. Ablation Studies

Longer Sequences: As one of the major differences between our method and the adapted NeRF is the different feature representation as input for the fine-level neural implicit function, we also tested how this impacts the generalization and fitting power of the approaches. To achieve that, we train our method as well as the temporal conditioned NeRF on sequences with variable length (1, 40, 120, 240, 360 frames) and report their reconstruction performance on a set of views at certain time frames. For all training sets, the first frame is shared and is taken as the test frame. For comparisons, we evaluate three different resolutions (16, 32, 64) for the coarse-level voxel feature in our method to better understand how the voxel resolution could affect the generalization capabilities and expressiveness of our model. Figure 4 shows the plot of MSE and SSIM v.s. the length of the training sequence of different models. A direct visual comparison between models trained on a different number of frames is shown in Figure 3. As can be seen, the performance of NeRF with a global animation code drops signif-

Figure 2. **Qualitative comparison of rendered images**. Our method recovers more fine-scale details than NV and NeRF, particularly in high-frequency regions like the eyes and hair. Results are rendered at 1024×667 with insets for better visualization.

	Sequence1			Sequence2			Sequence3			Sequence4		
	MSE	PSNR	SSIM	MSE	PSNR	SSIM	MSE	PSNR	SSIM	MSE	PSNR	SSIM
NV	46.19	31.56	0.8851	52.11	31.24	0.8499	83.07	29.24	0.7742	40.47	32.30	0.9086
NeRF	43.34	31.88	0.8923	46.89	31.79	0.8531	90.45	28.87	0.7727	35.52	32.95	0.9129
Ours	**34.01**	**33.09**	**0.9064**	**42.65**	**32.24**	**0.8617**	**79.29**	**29.61**	**0.7826**	**27.62**	**34.12**	**0.9246**

Table 1. **Image prediction error**. We compare NV, NeRF, and our method on 4 sequences, and report average error computed over a set of approximately 200 images of 7 views for each sequence. Our method outperforms all other baselines on all metrics.

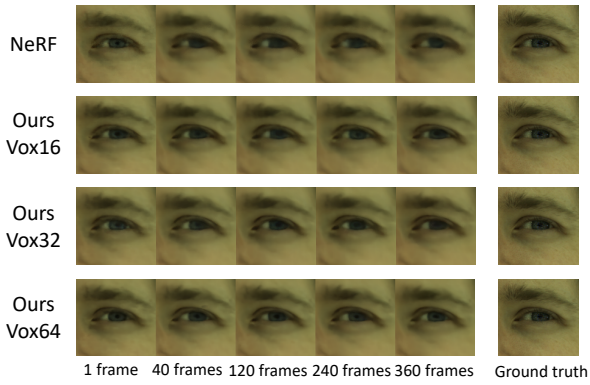

Figure 3. **Effect of sequence length on quality.** Conditioning the radiance field using local animation codes instead of a global code greatly expands model capacity, allowing our model to recover much sharper images even when trained on longer video sequences.

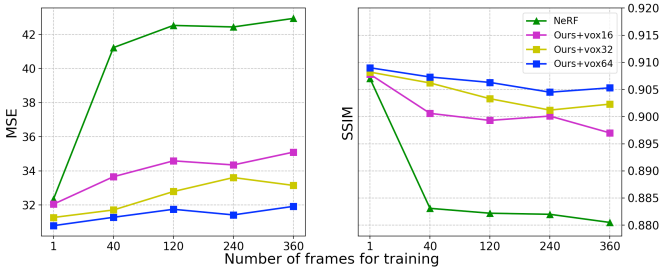

Figure 4. **Effect of sequence length on reconstruction.** MSE and SSIM on the first frame v.s. length of the training sequence.

icantly when the total number of training frames increases, while our method maintains a higher rendering quality due to a more expressive local voxel animation code, which enforces the fine-level implicit function to learn local rather than global representations and capture high frequency details more accurately. In addition, the 3D convolutional decoder imposes 3D inductive bias and improves the capacity of the whole model with the help of a 3D voxel feature tensor that has more spatial awareness compared to a global code. We also see that rendering quality improves and the model achieves better generalization when the coarse-level voxel feature resolution is relatively large. As can be seen in Fig. 3, when the resolution is smaller, the performance drops as each local code is responsible for describing a larger region of space.

	MSE	PSNR	SSIM	Runtime
NeRF+HS	36.33	32.90	0.8898	>25s
NeRF+SS	38.80	32.75	0.8886	19.69s
Ours+HS	**27.23**	**34.24**	0.9090	14.30s
Ours+SS	30.35	34.13	**0.9113**	**3.6s**

Table 2. **Ablation on different sampling schemes.** We show image reconstruction results as well as runtime for both NeRF and ours with different sampling strategies.

	MSE	PSNR	SSIM
keypoints encoder **w/o** ft	58.52	30.78	0.8891
image encoder **w/o** ft	55.86	31.07	0.8903
keypoints encoder **w/** ft	35.12	32.90	0.9024
image encoder **w/** ft	34.86	33.27	0.9053
full model ft	**32.47**	**33.84**	**0.9121**

Table 3. **Novel content synthesis.** We show results on novel content generation and novel sequence fitting. We tested two different encoder models that use data from two modalities: sparse 2D keypoints and images.

Sampling Strategy and Runtime Comparison: We further trained our method and NeRF on a single sentence applying different sampling schemes: the hierarchical sampling (HS) in [23] and our simple sampling (SS). We show results in Tab. 2. As we can see our simple sampling preserves rendering quality while enjoying a large increase in runtime efficiency. For rendering an image with resolution 1024×667, NV takes roughly 0.9s and NeRF is taking >25s whereas our methods takes 3.6s.

4.5. Animation

We demonstrate a large variety of applications that is enabled by our approach.

Latent Space Sampling and Interpolation Given the encoder-decoder architecture, we can generate smooth transitions between two expressions by interpolating in latent space and create free-view animations. In Fig. 5(b), we show direct interpolation results between two frames with different expressions. The frames in the red and blue bounding box are two key frames and all other frames inbetween are interpolated results. We also show rendering results by randomly sampling in the latent space in Fig. 5(c).

Landmark Driven Animation: Because the decoder only depends on a single global code to generate the dynamic

Figure 5. **Novel sequence generation**. New animations can be created by dynamically changing the global animation code, for example by (a) using keypoints to drive the animation, (b) interpolating the code at key frames, (c) sampling from the latent distribution, or (d) directly fitting the codes to match a novel sequence. Please refer to supplemental material for more visual results.

field, the original image encoder can be switched to an encoder that takes inputs from other modalities as long as correspondence between input and outputs can be established. To demonstrate controllable animation, we use 2d landmarks as a substitute of the image input and train a simplified PointNet [27]-like encoder that regresses the global code from the set of 2d landmarks. To train such an encoder, we minimize the ℓ_2 distance between the global code z_{kps} from keypoints and its corresponding global code z_{img} from the image on the training set. Fig. 5(a) shows some rendering results that are driven by a keypoint encoder. To test generalization to a novel sentence that is not included in the training data, we deployed the keypoint encoder and the pre-trained decoder on a novel sequence. Results on test views are reported in Tab. 3. We can see, that with a keypoint encoder using only a regression loss in the latent space, the avatar can be driven with reasonable performance, even though keypoints provide less information than images.

Novel Sequence Fitting: To demonstrate our model's ability to generalize to a novel sequence, we show results of animations driven by novel video sequences. For novel sequence generation from a given input modality, two components need to generalize: (1) the encoder, which produces animation codes given novel image inputs, and (2) the decoder, which renders novel animation codes into images. We first study the generalization ability of the decoder in isolation. To do this, we fine-tune the encoder on the novel sequence, fixing the parameters of the decoder and only back-propagating gradients to the encoder's parameters. Fig. 5(d) shows rendering results. To test the ability of generalization to novel input driving sequences, we test the complete encoder-decoder model on a novel sequence, without any fine-tuning. Results are shown in Tab. 3. As we can see, an image-based encoder trained with a photometric loss shows better performance on novel content than a key-point encoder trained with a regression loss on the latent space. Innately, image input is a more informative input than sparse key-points. Training with a photometric loss rather than a regression loss enables the encoder to output latent codes that are more compatible with the decoder. We also fine-tuned just the image encoder with a photometric loss to align the latent space and we find that the rendering results achieve compatible quality on novel content. We also fine-tuned the full model (both encoder and decoder) and we find the gap is not large in comparison to the model that only has its encoder fine-tuned.

5. Limitations

While we achieve state-of-the-art results, our approach is still subject to a few limitations which can be addressed in follow-up work: (1) Our method heavily relies on the quality of the coarse-level voxel field. In cases when the voxel representation has significant errors, the following fine-level model is likely not to recover. (2) Since we rely on the voxel field for our coarse-level representation, our method is primarily applicable to object-centric scenes. Potentially, by substituting the voxelized representation with a coarse depth map, it could also be applied to arbitrary scenes. (3) Although our compositional approach improves the scalability of both voxel-based and continuous representations, our approach is still limited in terms of resolution. One possibility to tackle this could be to also regress the positions and locations of the voxels or a group of voxels, which could serve as a more efficient and reliable proxy.

6. Conclusion

In this paper, we proposed a method for rendering and driving photo-realistic avatars of humans captured with a multi-view camera system. Our representation bridges the gap between discrete and continuous volumetric representations by combining a coarse 3D-structure-aware grid of animation codes with a continuous learned scene function that enables high-resolution detail without the need for a dense voxel grid. We show that our approach produces higher-quality results than previous methods, especially as the length of the sequence increases, and is significantly faster than classical neural radiance fields. Our approach also enables driving the model, which we demonstrate via

interpolation in the latent space, randomly sampling the latent space, and facial motion control via a set of sparse keypoints. We believe that our approach is a stepping stone towards higher-quality telepresence systems.

References

[1] Kara-Ali Aliev, Artem Sevastopolsky, Maria Kolos, Dmitry Ulyanov, and Victor Lempitsky. Neural point-based graphics. *arXiv preprint arXiv:1906.08240*, 2019. 1, 2

[2] Benjamin Attal, Selena Ling, Aaron Gokaslan, Christian Richardt, and James Tompkin. Matryodshka: Real-time 6dof video view synthesis using multi-sphere images. *arXiv preprint arXiv:2008.06534*, 2020. 2

[3] 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 Trans. Graph.*, 39(4), July 2020. 2

[4] Rohan Chabra, Jan Eric Lenssen, Eddy Ilg, Tanner Schmidt, Julian Straub, Steven Lovegrove, and Richard Newcombe. Deep local shapes: Learning local sdf priors for detailed 3d reconstruction. *arXiv preprint arXiv:2003.10983*, 2020. 3

[5] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 3

[6] Wenzheng Chen, Huan Ling, Jun Gao, Edward Smith, Jaakko Lehtinen, Alec Jacobson, and Sanja Fidler. Learning to predict 3d objects with an interpolation-based differentiable renderer. In *Advances in Neural Information Processing Systems*, pages 9609–9619, 2019. 2

[7] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European conference on computer vision*, pages 628–644. Springer, 2016. 2

[8] Kyle Genova, Forrester Cole, Aaron Maschinot, Aaron Sarna, Daniel Vlasic, and William T Freeman. Unsupervised training for 3d morphable model regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8377–8386, 2018. 1, 2

[9] Kyle Genova, Forrester Cole, Avneesh Sud, Aaron Sarna, and Thomas Funkhouser. Local deep implicit functions for 3d shape. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4857–4866, 2020. 3

[10] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, and Thomas Funkhouser. Local implicit grid representations for 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6001–6010, 2020. 3

[11] Abhishek Kar, Christian Häne, and Jitendra Malik. Learning a multi-view stereo machine. In *Advances in neural information processing systems*, pages 365–376, 2017. 2

[12] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[13] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 3, 5

[14] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields, 2020. 1, 3

[15] Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7708–7717, 2019. 1, 2

[16] Stephen Lombardi, Jason Saragih, Tomas Simon, and Yaser Sheikh. Deep appearance models for face rendering. *ACM Trans. Graph.*, 37(4), July 2018. 1

[17] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *ACM Trans. Graph.*, 38(4), July 2019. 1, 2, 5, 6

[18] Matthew M Loper and Michael J Black. Opendr: An approximate differentiable renderer. In *European Conference on Computer Vision*, pages 154–169. Springer, 2014. 2

[19] Feng Liu Xiaoming Liu Luan Tran. Towards high-fidelity nonlinear 3d face morphoable model. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, 2019. 1, 2

[20] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3

[21] Moustafa Meshry, Dan B Goldman, Sameh Khamis, Hugues Hoppe, Rohit Pandey, Noah Snavely, and Ricardo Martin-Brualla. Neural rerendering in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6878–6887, 2019. 1, 2

[22] Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, 2019. 1, 2

[23] 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. *arXiv preprint arXiv:2003.08934*, 2020. 1, 3, 4, 5, 6, 7

[24] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 3

[25] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019. 3

[26] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy

networks. In *European Conference on Computer Vision (ECCV)*, 2020. 3

[27] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 8

[28] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. *arXiv preprint arXiv:2007.02442*, 2020. 3, 4

[29] V. Sitzmann, J. Thies, F. Heide, M. Nießner, G. Wetzstein, and M. Zollhöfer. Deepvoxels: Learning persistent 3d feature embeddings. In *Proceedings of Computer Vision and Pattern Recognition (CVPR 2019)*, 2019. 1, 2

[30] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *Advances in Neural Information Processing Systems*, pages 1121–1132, 2019. 1, 3

[31] Ayush Tewari, Michael Zollhöfer, Pablo Garrido, Florian Bernard, Hyeongwoo Kim, Patrick Pérez, and Christian Theobalt. Self-supervised multi-level face model learning for monocular reconstruction at over 250 hz. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2

[32] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *ACM Transactions on Graphics 2019 (TOG)*, 2019. 1

[33] Shubham Tulsiani, Tinghui Zhou, Alexei A Efros, and Jitendra Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2626–2634, 2017. 2

[34] Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. Synsin: End-to-end view synthesis from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7467–7477, 2020. 1, 2

[35] Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Advances in neural information processing systems*, pages 82–90, 2016. 2

[36] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018. 1, 2