

DynIBaR: Neural Dynamic Image-Based Rendering

Zhengqi Li¹, Qianqian Wang^{1,2}, Forrester Cole¹, Richard Tucker¹, Noah Snavely¹

¹Google Research ²Cornell Tech

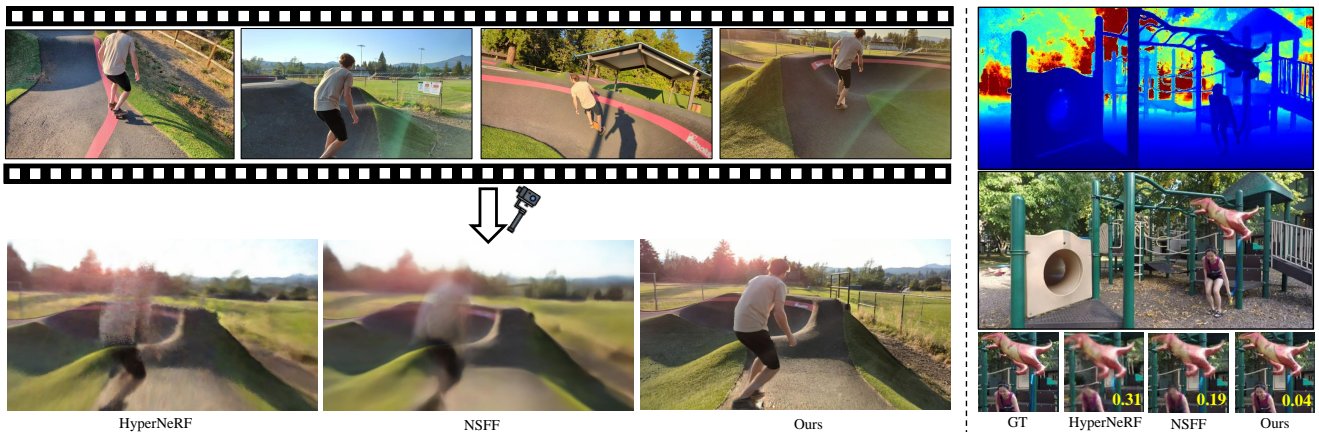


Figure 1. Recent methods for synthesizing novel views from monocular videos of dynamic scenes—like HyperNeRF [49] and NSFF [35]—struggle to render high-quality views from long videos featuring complex camera and scene motion. We present a new approach that addresses these limitations, illustrated above via an application to 6DoF video stabilization, where we apply our approach and prior methods on a 25 second, shaky video clip, and compare novel views rendered along a smoothed camera path (**left**). In this example, our approach produces much sharper results. On a dynamic scenes dataset (**right**) [75], our approach significantly improves rendering fidelity, as indicated by synthesized images and LPIPS errors computed on pixels corresponding to moving objects (yellow numbers). Please see the supplementary video for full results.

Abstract

We address the problem of synthesizing novel views from a monocular video depicting a complex dynamic scene. State-of-the-art methods based on temporally varying Neural Radiance Fields (aka dynamic NeRFs) have shown impressive results on this task. However, for long videos with complex object motions and uncontrolled camera trajectories, these methods can produce blurry or inaccurate renderings, hampering their use in real-world applications. Instead of encoding the entire dynamic scene within the weights of an MLP, we present a new approach that addresses these limitations by adopting a volumetric image-based rendering framework that synthesizes new viewpoints by aggregating features from nearby views in a scene-motion-aware manner. Our system retains the advantages of prior methods in its ability to model complex scenes and view-dependent effects, but also enables synthesizing photo-realistic novel views from long videos featuring complex scene dynamics with

unconstrained camera trajectories. We demonstrate significant improvements over state-of-the-art methods on dynamic scene datasets, and also apply our approach to in-the-wild videos with challenging camera and object motion, where prior methods fail to produce high-quality renderings.

1. Introduction

Computer vision methods can now produce reconstructions of static 3D scenes that allow for renderings with spectacular quality. What about moving scenes, like those featuring people or pets? Novel view synthesis from a monocular video of dynamic scene is a much more challenging dynamic scene reconstruction problem. Recent work has made great progress towards synthesis of novel views in both space and time thanks to the emergence of time-varying neural volumetric representations, such as HyperNeRF [49] and Neural Scene Flow Fields (NSFF) [35], which encode spatiotemporally varying scene content volumetrically as the weights of

coordinate-based multi-layer perceptrons (MLPs).

However, these *dynamic NeRF* methods have fundamental limitations that prevent them from applying to casual, in-the-wild videos. Local scene flow-based methods such as NSFF struggle to scale to longer input videos captured with unconstrained camera motions: the NSFF paper only claims good performance for 1-second, forward-facing videos [35]. Methods like HyperNeRF that construct a canonical model can handle long sequences, but are mostly constrained to object-centric scenes with controlled camera paths, and can fail on dynamic scenes with complex object motion.

In this work, we present a novel approach that is scalable to dynamic scenes captured with 1) long time duration, 2) unconstrained scene configurations, 3) uncontrolled camera trajectories, and 4) fast and complex object motion. Our approach retains the advantages of volumetric scene representations that can model intricate scene geometry with view-dependent effects, while significantly improving rendering fidelity of both static and dynamic scene contents compared to recent methods [35, 49], as illustrated in Fig. 1.

We take inspiration from recent work on rendering of static scenes in which images are synthesized by aggregating local image features along epipolar lines from nearby views through ray transformers [38, 63, 70]. These methods show promise in scaling to longer inputs and larger camera motions. However, scenes that are in motion violate the epipolar constraints assumed by those methods. We propose to aggregate multi-view image features in *scene motion-adjusted* ray space, which allows us to correctly reason about spatio-temporally varying geometry and appearance.

We also observed many efficiency and robustness challenges in scaling aggregation-based methods up to dynamic scenes. To efficiently model scene motion across multiple views, we model this motion using *motion trajectory fields* that span multiple frames, represented with learned basis functions. Furthermore, to achieve temporal coherence of the dynamic scene reconstruction, we introduce a new temporal photometric loss that operates in motion-adjusted ray space. Lastly, to improve rendering quality in space-time view synthesis, we propose to factor the scene into static and dynamic components through a new IBR-based motion segmentation technique within a Bayesian learning framework.

On two dynamic scene benchmarks, we show that our approach enables rendering highly detailed scene contents, and significantly improves upon the state-of-the-art, leading to an average reduction in LPIPS errors by over 50% on regions corresponding to moving objects and across entire scenes. We also show that our method can be applied to in-the-wild videos with long durations, complex scene motion, and varied camera trajectories, where prior state-of-the-art methods fail to produce high quality renderings. We hope that our work advances the ability of dynamic view synthesis techniques to work on in-the-wild videos.

2. Related Work

Novel view synthesis. Image-based rendering synthesizes novel views by integrating pixel information from input images [57] and can be classified into different categories depending on how much it relies on geometry. Light field or lumigraph rendering [9, 20, 26, 32] generate new views by filtering and interpolating ray samples, without requiring explicit geometric models. To handle sparser input views, many approaches [7, 14, 18, 22, 23, 26, 30, 51, 53, 54] leverage pre-computed proxy geometry such as depth maps or meshes for rendering novel views.

Recently, neural representations [12, 17, 37, 39, 45, 47, 58–61, 72, 81] have demonstrated great potential for high-quality novel view synthesis. In particular, Neural Radiance Fields (NeRF) [45] achieves an unprecedented level of fidelity by representing continuous scene radiance fields encoded as multi-layer perceptrons (MLPs). Among all methods building on NeRF, IBRNet [70] is the most relevant to our work. IBRNet integrates classical image-based rendering techniques with volume rendering to achieve generalization to unseen scenes. Our work extends this kind of volumetric image-based rendering framework designed for static scenes [11, 63, 70] to more challenging dynamic scenes. Note that our focus is on modeling long videos with complex camera and object motion while synthesizing higher-quality novel views, rather than on generalization across scenes.

Dynamic scene view synthesis. Our method is related to geometry reconstruction of dynamic scenes from RGBD [5, 15, 24, 46, 68, 83] or monocular videos [31, 43, 78, 80]. However, depth- or mesh-based representations have difficulty modeling complex geometry and view-dependent effects.

Most prior work on novel view synthesis for dynamic scenes requires synchronized multi-view videos as input [1, 3, 6, 27, 33, 62, 69, 76, 82], which limits their real world applicability. Some methods [8, 13, 21, 50, 71] use domain knowledge such as template models to achieve high-quality synthesis results, but are restricted to specific categories [40, 55]. More recently, many works have devised methods for synthesizing novel views of a dynamic scene from a single camera. Yoon *et al.* [75] propose to render novel views by performing explicit warping using depth maps, which are obtained via single-view depth and multi-view stereo. However, this method fails to model complex scene geometry and fill in realistic and consistent content at disocclusion. With advances in neural rendering, NeRF-based dynamic view synthesis methods have shown state-of-the-art results [16, 35, 52, 66, 74]. Many approaches, such as Nerfies [48] and HyperNeRF [49], represent scenes using a deformation field mapping each local observation to a canonical scene representation. These deformations are conditioned on time [52] or a per-frame latent code [48, 49, 66], and are parameterized as translations [52, 66] or rigid body

motion fields [48, 49]. These methods can handle long videos, but are largely limited to object-centric scenes with fairly small object motion and controlled camera paths. Another class of methods represent the scene as time-varying NeRFs [19, 35, 67, 74]. In particular, NSFF uses neural scene flow fields that can capture fast and complex 3D scene motion for in-the-wild videos [35]. However, this method only works well for forward-facing videos of 1-2 seconds.

3. Dynamic Image-Based Rendering

Given a monocular video of a dynamic scene, with frames (I_1, I_2, \dots, I_N) and known camera parameters $(\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_N)$, our goal is to synthesize a novel view-point at any desired time within the video. As in many other approaches, we train per-video: first optimizing our models to reconstruct the input frames, and then using these models to render novel views.

Rather than encoding 3D color and density directly in the weights of an MLP as in recent dynamic NeRF methods, we integrate ideas from classical IBR techniques within a volumetric rendering framework. Compared to explicit surfaces, volumetric representations can more readily model complex scene geometry with view-dependent effects.

The following sections introduce our methods for scene-motion-adjusted multi-view feature aggregation (Sec. 3.1), and enforcing temporal consistency via *cross-time rendering in motion-adjusted ray space* (Sec. 3.2). Our full system combines a static model and a dynamic model to produce a color at each pixel. Accurate scene factorization is achieved by bootstrapping using segmentation masks derived from a separately trained motion segmentation module within a Bayesian learning framework. (Sec. 3.3).

3.1. Motion-adjusted feature aggregation

Multi-view feature aggregation. We synthesize new views using features extracted from temporally nearby source views. To render an image at time i , we first identify the source views I_j within r frames of i , that is, where $j \in \mathcal{N}(i) = [i - r, i + r]$. For each source view, we extract a 2D feature map F_i through a shared convolutional encoder network to form an input tuple $\{I_j, \mathbf{P}_j, F_j\}$.

To predict the color and density of each point sampled along a target ray \mathbf{r} , we must aggregate source view features while accounting for scene motion. For a static scene, points along a target ray will always lie along a corresponding epipolar line in a neighboring source view, hence potential correspondences are easily determined and combined by sampling pixel locations along these neighboring epipolar lines [63, 70]. However, moving scene elements violate epipolar constraints, leading to inconsistent feature aggregation if motion is not taken into account. To determine correspondence in dynamic scenes, one straightforward idea

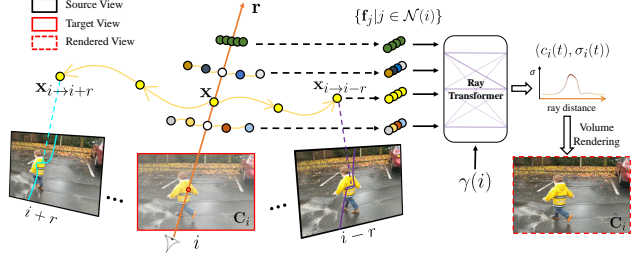


Figure 2. **Rendering via motion-adjusted multi-view feature aggregation.** Given a sampled location \mathbf{x} at time i along a target ray \mathbf{r} , we estimate its motion trajectory, which determines the 3D correspondence of \mathbf{x} at nearby time $j \in \mathcal{N}(i)$, denoted $\mathbf{x}_{i \rightarrow j}$. Each warped point is then projected into its corresponding source view. Image features \mathbf{f}_j extracted along the projected curves are aggregated and fed to the ray transformer with time embedding $\gamma(i)$, producing per-sample color and density (\mathbf{c}_i, σ_i) . The final pixel color $\hat{\mathbf{C}}_i$ is then synthesized by volume rendering (\mathbf{c}_i, σ_i) along \mathbf{r} .

is to estimate a scene flow field via an MLP [35] to determine a given point’s motion-adjusted 3D location at a nearby time, as shown in the second column of Fig. 4. However, this strategy is computational infeasible in a volumetric IBR framework due to recursive unrolling of the MLPs.

Motion trajectory fields. Instead, we represent scene motion using motion trajectory fields with learned basis functions. In particular, for a given point \mathbf{x} along target ray \mathbf{r} at time i , we encode its trajectory coefficients with a MLP $G_{\text{MT}}: \{\phi_i^l(\mathbf{x})\}_{l=1}^L = G_{\text{MT}}(\gamma(\mathbf{x}), \gamma(i))$, where $\phi_i^l \in \mathcal{R}^3$, and $L = 2r$. γ denotes positional encoding with linearly increasing frequency, where we set the number of frequencies to 16 based on the assumption that scene motion tends to be low frequency [80].

In addition, we assign a learnable basis $\{h_i^l\}_{l=1}^L, h_i^l \in \mathcal{R}$ to every input time step i , which is optimized jointly with the MLP. The motion trajectory of \mathbf{x} is then defined as $\Gamma_{\mathbf{x}, i}(j) = \sum_{l=1}^L h_j^l \phi_i^l(\mathbf{x})$, and thus, the relative displacement between \mathbf{x} and its 3D correspondence $\mathbf{x}_{i \rightarrow j}$ at time j is computed as

$$\Delta_{\mathbf{x}, i}(j) = \Gamma_{\mathbf{x}, i}(j) - \Gamma_{\mathbf{x}, i}(i). \quad (1)$$

With this motion trajectory representation, finding 3D correspondences for a query point \mathbf{x} in neighboring views requires just a single MLP query, allowing efficient multi-view feature aggregation within our volume rendering framework. We initialize the basis $\{h_i^l\}_{l=1}^L$ with the DCT basis as proposed by Wang *et al.* [67], but fine-tune the basis with other modules during optimization, since we observe that a fixed DCT basis can fail to model a wide range of real-world motions, as shown in the third column of Fig. 4.

Using the estimated motion trajectory of \mathbf{x} at time i , we denote \mathbf{x} ’s corresponding 3D point at time j as $\mathbf{x}_{i \rightarrow j} = \mathbf{x} + \Delta_{\mathbf{x}, i}(j)$. We project each warped point $\mathbf{x}_{i \rightarrow j}$ into its source view I_j using camera parameters \mathbf{P}_j , and extract

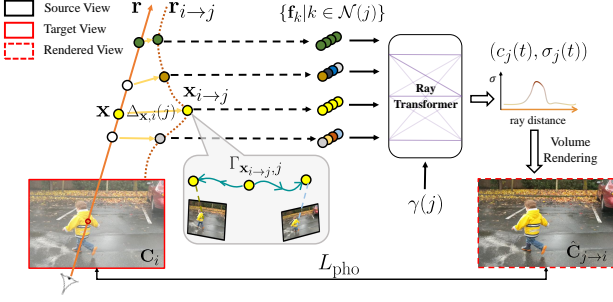


Figure 3. **Temporal consistency via cross-time rendering.** To enforce temporal consistency in a dynamic reconstruction, we render each frame I_i using the scene model from a nearby time j , which we call *cross-time rendering*. A ray \mathbf{r} from image i is instead rendered using a curved ray $\mathbf{r}_{i \rightarrow j}$, i.e., \mathbf{r} warped to time j . That is, having computed the motion-adjusted point $\mathbf{x}_{i \rightarrow j} = \mathbf{x} + \Delta_{\mathbf{x}, i}(j)$ at nearby time j from every sampled location along \mathbf{r} , we query $\mathbf{x}_{i \rightarrow j}$ and time j via the MLP to predict its motion trajectory $\Gamma_{\mathbf{x}_{i \rightarrow j}, j}$, along which we aggregate image features \mathbf{f}_k extracted from source views within time $k \in \mathcal{N}(j)$. The aggregated features along $\mathbf{r}_{i \rightarrow j}$ are fed to the ray transformer with time embedding $\gamma(j)$ to produce per-sample color and density (c_j, σ_j) at time j . A pixel color $\hat{C}_{j \rightarrow i}$ is computed by volume rendering (c_j, σ_j) , and then compared to the ground truth color C_i to form a reconstruction loss \mathcal{L}_{pho} .

color and feature vectors \mathbf{f}_j at the projected 2D pixel locations. The resulting set of source features is fed to a shared MLP whose output features are aggregated through weighted average pooling [70] to produce a single feature vector at each sample point location. A ray transformer network with time embedding $\gamma(i)$ then processes the sequence of aggregated features along ray \mathbf{r} to predict per-sample colors and densities (c_i, σ_i) , as shown in Fig. 2. We then use standard NeRF volume rendering [4] to obtain the final pixel color $\hat{C}_i(\mathbf{r})$ for ray \mathbf{r} from this sequence of colors and densities.

3.2. Cross-time rendering for temporal consistency

If we optimize our dynamic scene representation by comparing \hat{C}_i with C_i alone, the representation might overfit to the input images, in that it might perfectly reconstruct those views, but fail to render correct novel views. This can happen because, for instance, the representation has the power to reconstruct completely separate models for each input time instance, without utilizing or accurately reconstructing scene motion. Therefore, to recover a consistent scene with physically plausible motion, we enforce temporal coherence of the scene representation. One way to define temporal coherence in this context is that the scene at two neighboring times i and j should be consistent when taking scene motion into account [19, 35, 67].

In addition, there is another key issue we face when optimizing the scene representation for an input video. To render an input view at time i during the optimization phase, we

gather local features from nearby times $j \in \mathcal{N}(i)$ and aggregate them using the ray transformer. However, if we include the target image feature \mathbf{f}_i itself in this aggregation, this will lead to a trivial solution where the model only uses \mathbf{f}_i to reconstruct I_i . On the other hand, at render time, excluding \mathbf{f}_i from the aggregation is a disadvantage when generating novel views at time i , because we wish to use as much information as possible from the target time itself to recover dynamic scene content.

To address both of these issues, we enforce temporal photometric consistency in our optimized representation via *cross-time rendering in motion-adjusted ray space*, as shown in Fig. 3. The idea is to render a view at time i but via some nearby time j , which we refer to as cross-time rendering. For each nearby time $j \in \mathcal{N}(i)$, instead of directly using points \mathbf{x} along ray \mathbf{r} , we consider the points $\mathbf{x}_{i \rightarrow j}$ along motion-adjusted ray $\mathbf{r}_{i \rightarrow j}$ and treat them as if they lie along a ray at time j .

Specifically, having computed the motion-adjusted points $\mathbf{x}_{i \rightarrow j}$, we query the MLP to predict the coefficients of *new* trajectories $\{\phi_j^l(\mathbf{x}_{i \rightarrow j})\}_{l=1}^L = G_{\text{MT}}(\mathbf{x}_{i \rightarrow j}, \gamma(j))$, and use these to compute corresponding 3D points $(\mathbf{x}_{i \rightarrow j})_{j \rightarrow k}$ for images k in the temporal window $\mathcal{N}(j)$, using Eq. 1. These new 3D correspondences are then used to render a pixel color exactly as described for a “straight” ray \mathbf{r}_i in Sec. 3.1, except now along the curved, motion-adjusted ray $\mathbf{r}_{i \rightarrow j}$. That is, each point $(\mathbf{x}_{i \rightarrow j})_{j \rightarrow k}$ is projected into its source view I_k and feature maps F_k with camera parameters \mathbf{P}_k to extract an RGB color and image feature \mathbf{f}_k , and then these features are aggregated and input to the ray transformer with the time embedding $\gamma(j)$. The result is a sequence of colors and densities (c_j, σ_j) along $\mathbf{r}_{i \rightarrow j}$ at time j , which can be composited through volume rendering to form a color $\hat{C}_{j \rightarrow i}$.

We can then compare $\hat{C}_{j \rightarrow i}(\mathbf{r})$ with the target pixel $C_i(\mathbf{r})$ via a motion-disocclusion-aware weighted RGB reconstruction loss:

$$\mathcal{L}_{\text{pho}} = \sum_{\mathbf{r}} \sum_{j \in \mathcal{N}(i)} \hat{\mathbf{W}}_{j \rightarrow i}(\mathbf{r}) \rho(C_i(\mathbf{r}), \hat{C}_{j \rightarrow i}(\mathbf{r})). \quad (2)$$

We use a generalized Charbonnier loss [10] for the RGB loss ρ . $\hat{\mathbf{W}}_{j \rightarrow i}(\mathbf{r})$ is a motion disocclusion weight computed by the difference of accumulated alpha weights between time i and j to address the motion disocclusion ambiguity described by NSFF [35] (see supplement for more details). Note that when $j = i$, there is no scene motion-induced displacement, meaning $\hat{C}_{j \rightarrow i} = \hat{C}_i$ and no disocclusion weights are involved ($\hat{\mathbf{W}}_{j \rightarrow i} = 1$).

Notably, when we render a target pixel at time i with 3D scene content borrowed from nearby time j , we can use features from time j as input to the ray transformer with time embedding $\gamma(j)$ without leading to a trivial solution, since the target pixel we compare to is at different time i . We show a comparison between our method with and without enforcing temporal consistency in the first column of Fig. 4.

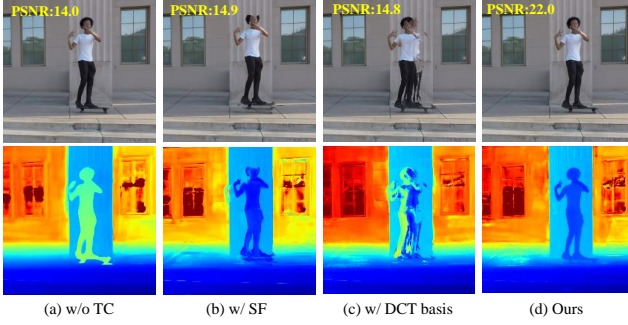


Figure 4. **Qualitative ablations.** From left to right, we show rendered novel views (top) and depths (bottom) from our system (a) without enforcing temporal consistency, (b) aggregating image features with scene flow fields instead of motion trajectories, (c) representing motion trajectory with a fixed DCT basis instead of a learned one, and (d) with full configuration. Simpler configurations significantly degrade reconstruction quality as indicated by PSNR calculated over the regions of moving objects.

3.3. Combining static and dynamic models

As observed in NSFF, estimating scene content using a small temporal window is insufficient to recover complete and high-quality content for static regions for novel view synthesis, since such content may not be observed in temporally nearby frames. Therefore, we follow the ideas of NSFF [35], and model the entire scene using two separate representations. Dynamic content (\mathbf{c}_i, σ_i) is represented with a time-varying model as above (used for cross-time rendering during optimization). Static content (\mathbf{c}, σ) is represented in the same way as the time-varying model but aggregates features sampled from nearby views without any motion adjustment (i.e., along epipolar lines), and without the additional time input to the ray transformer.

The dynamic and static predictions are combined and rendered to a single output color $\hat{\mathbf{C}}_i^{\text{full}}$ (or $\hat{\mathbf{C}}_{j \rightarrow i}^{\text{full}}$ through cross-time rendering) using the method for combining static and transient models of NeRF-W [44]. Each model’s color and density estimates can also be rendered separately, giving color $\hat{\mathbf{C}}^{\text{st}}$ for static content and $\hat{\mathbf{C}}_i^{\text{dy}}$ for dynamic content.

When combining the two representations, we rewrite the photometric consistency term in Eq. 2 as a loss comparing $\hat{\mathbf{C}}_{j \rightarrow i}^{\text{full}}(\mathbf{r})$ with target pixel $\mathbf{C}_i(\mathbf{r})$:

$$\mathcal{L}_{\text{pho}} = \sum_{\mathbf{r}} \sum_{j \in \mathcal{N}(i)} \hat{\mathbf{W}}_{j \rightarrow i}(\mathbf{r}) \rho(\mathbf{C}_i(\mathbf{r}), \hat{\mathbf{C}}_{j \rightarrow i}^{\text{full}}(\mathbf{r})) \quad (3)$$

Motion segmentation via hybrid IBR. In our volumetric IBR framework, we observed that without any initialization, scene factorization tends to be dominated by either the time-invariant or the time-varying representation, a phenomena also observed in recent methods [28, 41]. To facilitate factorization, Gao *et al.* [19] initialize the systems by adopt-

ing masks from semantic segmentation, relying on the assumption that all moving objects are captured by a set of candidate semantic segmentation labels, and segmentation masks are temporally accurate. However, these assumptions do not hold in many real-world scenarios as observed by Zhou *et al.* [79]. Instead, we propose an IBR-based motion segmentation module that produces segmentation masks for bootstrapping our main two-component scene representations. Our idea is inspired by the Bayesian learning techniques proposed in recent work [44, 79], but integrated into a volumetric IBR representation.

In particular, before training our main two-component scene representation, we jointly train two lightweight models to obtain a motion segmentation mask M_i for each input frame I_i . We model static scene content with an IBR-Net [70] that renders a pixel color $\hat{\mathbf{B}}^{\text{st}}$ using volume rendering along each ray via feature aggregation along epipolar lines from nearby source views without considering scene motion. We model dynamic scene content with a 2D convolutional encoder-decoder network D , which predicts a 2D opacity map α_i^{dy} , confidence map β_i^{dy} , and RGB image $\hat{\mathbf{B}}_i^{\text{dy}}$ from an input frame:

$$\hat{\mathbf{B}}_i^{\text{dy}}, \alpha_i^{\text{dy}}, \beta_i^{\text{dy}} = D(I_i). \quad (4)$$

The full reconstructed image is then composited pixelwise from the outputs of the two models:

$$\hat{\mathbf{B}}_i^{\text{full}}(\mathbf{r}) = \alpha_i^{\text{dy}}(\mathbf{r}) \hat{\mathbf{B}}_i^{\text{dy}}(\mathbf{r}) + (1 - \alpha_i^{\text{dy}}(\mathbf{r})) \hat{\mathbf{B}}^{\text{st}}(\mathbf{r}). \quad (5)$$

To effectively segment moving objects, we assume the observed pixel color is uncertain in a heteroscedastic aleatoric manner, and model the observed pixel color \mathbf{C}_i in the video with a Cauchy distribution with time dependent confidence β_i^{dy} . By taking the negative log-likelihood of \mathbf{C}_i , our segmentation loss can be written as a confidence-weighted reconstruction loss:

$$\mathcal{L}_{\text{seg}} = \sum_{\mathbf{r}} \log \left(\beta_i^{\text{dy}}(\mathbf{r}) + \frac{\|\hat{\mathbf{B}}_i^{\text{full}}(\mathbf{r}) - \mathbf{C}_i(\mathbf{r})\|^2}{\beta_i^{\text{dy}}(\mathbf{r})} \right). \quad (6)$$

By optimizing the two models using Eq. 6, we obtain a motion segmentation mask M_i by thresholding α_i^{dy} with 0.9. We show our estimated motion segmentation masks overlaid on input images in Fig. 5.

Bootstrapping with segmentation masks. After the motion segmentation stage, we bootstrap our main time-varying and time-invariant models with the masks M_i , as in Omnimate [42], by applying a reconstruction loss on renderings from the time-varying model in dynamic regions, and on renderings from the time-invariant model in static regions:

$$\begin{aligned} \mathcal{L}_{\text{mask}} = & \sum_{\mathbf{r}} (1 - M_i)(\mathbf{r}) \rho(\hat{\mathbf{C}}^{\text{st}}(\mathbf{r}), \mathbf{C}_i(\mathbf{r})) \\ & + \sum_{\mathbf{r}} M_i(\mathbf{r}) \rho(\hat{\mathbf{C}}_i^{\text{dy}}(\mathbf{r}), \mathbf{C}_i(\mathbf{r})) \end{aligned} \quad (7)$$

We initialize the system with $\mathcal{L}_{\text{mask}}$ and decay the weights by a factor of 10 and 2 for dynamic and static parts respectively every 50K optimization steps.

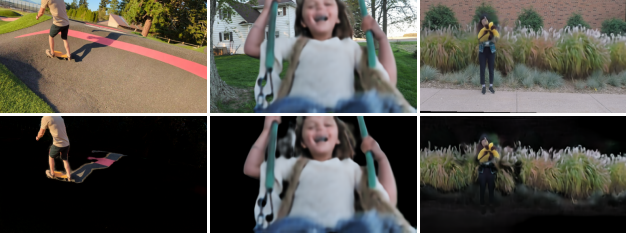


Figure 5. **Motion segmentation.** We show full rendering $\hat{\mathbf{B}}_i^{\text{full}}$ (top) and motion segmentation overlaid with rendered dynamic contents $\alpha_i^{\text{dy}} \odot \hat{\mathbf{B}}_i^{\text{dy}}$ (bottom). Our approach segments challenging dynamic elements such as the moving shadow and swing, as well as the swaying bushes.

3.4. Global spatial coordinate embedding

With local image feature aggregation alone, it is hard to determine density accurately on non-surface or occluded surface points due to inconsistent features from different source views, as described in NeuRay [38]. Therefore, to improve global reasoning for density prediction, we append a global spatial coordinate embedding as an input to the ray transformer, in addition to the time embedding [63].

For the time-invariant model, we concatenate each extracted local image feature at each sampled point with its corresponding (1) ray coordinates with respect to the target view, (2) ray coordinates with respect to the source view, and (3) xyz coordinates in the global reference frame. We use Plücker coordinates [25] to represent ray coordinates since these allow us to model arbitrary scene and camera geometry without ambiguities or singularities.

For the time-varying model, recall that local image features are aggregated along curved rays during the optimization stage as shown in Fig. 3, but once optimization is complete, novel views are rendered by aggregating features along straight rays cast directly from the desired target view. As a result, we observe that concatenating the 3D coordinates of \mathbf{x} or $\mathbf{x}_{i \rightarrow j}$ with the corresponding 2D image features as inputs to the ray transformer impairs view interpolation. Therefore, we concatenate the encoded global 3D coordinates of \mathbf{x} with the features output from the attention module in the ray transformer before predicting density and color (please see the supplement for more details of this design choice). For all spatial coordinates, we apply a Fourier features-based positional encoding [45, 64] with four frequencies.

3.5. Regularization

As noted in prior work, monocular reconstruction of complex dynamic scenes is highly ill-posed, and using photometric consistency alone is insufficient to avoid bad local minima during optimization [19, 35]. Therefore, we adopt regularization schemes used in prior work [2, 35, 73, 75], which consist of three main parts $\mathcal{L}_{\text{reg}} = \mathcal{L}_{\text{data}} + \mathcal{L}_{\text{MT}} + \mathcal{L}_{\text{cpt}}$. $\mathcal{L}_{\text{data}}$ is a data-

driven term consisting of ℓ_1 monocular depth and optical flow consistency priors from the estimates using Zhang *et al.* [80] and RAFT [65]. \mathcal{L}_{MT} is motion trajectory regularization term that encourages the estimated motion trajectory fields to be cycle consistent and spatial-temporally smooth; \mathcal{L}_{cpt} is a compactness prior that encourages the scene decomposition to be binary, mitigating floaters through entropy and distortion losses [2]. We refer readers to the supplement for more details.

In summary, the final combined loss to optimize our main representations for space-time view synthesis is as follows:

$$\mathcal{L} = \mathcal{L}_{\text{pho}} + \mathcal{L}_{\text{mask}} + \mathcal{L}_{\text{reg}}. \quad (8)$$

4. Evaluation

4.1. Implementation details

Datasets. We conduct numerical evaluations on the Nvidia Dynamic Scene Dataset [75] and UCSD Dynamic Scenes Dataset [36]. Each dataset consists of eight forward-facing dynamic scenes recorded by synchronized multi-view cameras. We follow prior work [35] to derive a monocular video from each sequence, where each video contains 100~250 frames. We removed frames that lack large regions of moving objects. We follow the protocol from prior work [35] that uses held-out images per time instance for evaluation.

View selection. For the time-varying, dynamic model we use a frame window radius of $r = 3$ for all experiments. For the time-invariant model representing static scene content, we use separate strategies for the dynamic scenes benchmarks and for in-the-wild videos. For the benchmarks, where camera viewpoints are located at discrete camera rig locations, we choose all nearby distinct viewpoints whose timestep is within 12 frames of the target time. For in-the-wild videos, naively choosing the nearest source views can lead to poor reconstruction due to insufficient camera baseline. Thus, to ensure that for any rendered pixel we have sufficient source views for computing its color, we select source views from distant frames. If we wish to select N^{vs} source views for the time-invariant model, we sub-sample every $\frac{2r^{\text{max}}}{N^{\text{vs}}}$ frames from the input video to build a candidate pool, where for a given target time i , we only search source views within $[i - r^{\text{max}}, i + r^{\text{max}}]$ frames. We estimate r^{max} using the method of Li *et al.* [34] based on SfM point co-visibility and camera relative baseline. We then construct a final set of source views for the model by choosing the top N^{vs} frames in the candidate pool that are the closest to the target view in terms of camera baseline. We set $N^{\text{vs}} = 16$.

Time interpolation. Our approach also allows for time interpolation by performing scene motion-based splatting, as introduced by NSFF [35]. To render at a specified target fractional time, we predict the volumetric density and color at two nearby input times by aggregating local image features from their corresponding set of source views. The predicted

Methods	Full			Dynamic Only		
	SSIM \uparrow	PSNR \uparrow	LPIPS \downarrow	SSIM \uparrow	PSNR \uparrow	LPIPS \downarrow
Nerfies [48]	0.609	20.64	0.204	0.455	17.35	0.258
HyperNeRF [49]	0.654	20.90	0.182	0.446	17.56	0.242
DVS [19]	0.913	26.05	0.081	0.778	22.63	0.144
NSFF [35]	<u>0.927</u>	<u>28.90</u>	<u>0.062</u>	<u>0.783</u>	<u>23.08</u>	0.159
Ours	0.958	30.92	0.027	0.827	24.32	0.061

Table 1. Quantitative evaluation on the Nvidia dataset [75].



Figure 6. Qualitative comparisons on the Nvidia dataset [75].

color and density are then splatted and linearly blended via the scene flow derived from our motion trajectories, and weighted according to the target fractional time index.

Setup. We estimate camera poses using COLMAP [56]. For each ray, we use a coarse-to-fine sampling strategy with 128 per-ray samples as in Wang *et al.* [70]. A separate model is trained for each scene using the Adam optimizer [29]. The network architecture for two main representations is a variant of that of IBRNet [70]. We reconstruct the entire scene in standard Euclidean space. Optimizing a full system on a 10 second video takes around two days using eight Nvidia A100s, and rendering takes roughly 20 seconds for a 768×432 frame. We refer readers to the supplemental material for our network architectures, hyper-parameters setting, and additional implementation details.

4.2. Baselines and error metrics

We compare our approach to state-of-the-art monocular view synthesis methods. Specifically, we compare to two recent canonical space-based methods, Nerfies [48] and HyperNeRF [49], and to two scene flow-based methods, NSFF [35] and Dynamic View Synthesis (DVS) from Gao *et al.* [19]. For fair comparisons, we use the same depth, optical flow and motion segmentation masks used for our approach as inputs to other methods.

Following prior work [35], we report the rendering quality of each method with three standard error metrics: peak signal-to-noise ratio (PSNR), structural similarity (SSIM), and perceptual similarity via LPIPS [77], and calculate errors both over the entire scene (Full) and restricted to moving regions (Dynamic Only).

Methods	Full			Dynamic Only		
	SSIM \uparrow	PSNR \uparrow	LPIPS \downarrow	SSIM \uparrow	PSNR \uparrow	LPIPS \downarrow
Nerfies [48]	0.823	24.32	0.096	0.595	18.45	0.234
HyperNeRF [49]	0.859	25.10	0.095	0.618	19.26	0.212
DVS [19]	0.923	29.57	0.089	<u>0.866</u>	<u>26.57</u>	<u>0.096</u>
NSFF [35]	<u>0.952</u>	<u>31.75</u>	<u>0.034</u>	0.851	25.83	0.115
Ours	0.983	36.48	0.014	0.909	28.02	0.042

Table 2. Quantitative evaluation on the UCSD dataset [36].



Figure 7. Qualitative comparisons on the UCSD dataset [36].

4.3. Quantitative evaluation

Quantitative results on the two benchmark datasets are shown in Table 1 and Table 2. Our approach significantly improves over prior state-of-the-art methods in terms of all error metrics. Notably, our approach improves PSNR over entire scene upon the second best methods by more than 2dB and 4dB on each of the two datasets. Our approach also reduces LPIPS error, a major indicator of perceptual quality compared with real images [77], by over 50%. These results suggest that our framework is much more effective at recovering highly detailed scene contents.

Ablation study. We conduct an ablation study on the Nvidia Dynamic Scene Dataset to validate the effectiveness of our various proposed system components. We show comparisons between our full system and variants in Tab. 3: A) baseline IBRNet [70] with extra time embedding; B) without enforcing temporal consistency via cross-time rendering; C) using scene flow fields to aggregate image features within one time step; D) without global spatial coordinate embedding; E) without using a time-invariant static scene model; F) without masked reconstruction loss via estimated motion segmentation masks; and G) without regularization loss. For this ablation study, we train each model more coarsely with 64 samples per ray. Without our motion trajectory representation and temporal consistency, view synthesis quality degrades significantly as shown in the first three rows of Tab. 3. Integrating a global spatial coordinate embedding further improves rendering quality. Combining static and dynamic models improves quality for static elements, as seen in metrics over full scenes. Finally, removing bootstrapping via motion segmentation masks or regularization reduces



Figure 8. **Qualitative comparisons on in-the-wild videos.** We show results on several 10-second videos of complex dynamic scenes featuring different types of camera trajectory. The leftmost column shows the start and end frames of each video; on the right we show novel views at intermediate times rendered from our approach and prior state-of-the-art methods [19, 35, 49].

performance, demonstrating the value of these losses for avoiding bad local minimal during optimization.

4.4. Qualitative evaluation

Dynamic scenes dataset. We provide qualitative comparisons between our approach and three prior state-of-the-art methods [19, 35, 49] on the two datasets in Fig. 6 and Fig. 7. Prior dynamic-NeRF methods have difficulty rendering details of moving objects, as seen in the excessively blurred dynamic content including the texture of balloons, human faces, and clothing. In contrast, our approach synthesizes photo-realistic novel views of both static and dynamic scene content and which are closest to the ground truth images.

In-the-wild videos. We also show qualitative comparisons on in-the-wild videos of complex dynamic scenes. Each video is 10 seconds (300 frames) long, and the videos feature different types of camera trajectories, including forward, backward and panning motions, and depict complex scenes with challenging object motions such as running and dancing, as shown in Fig. 8. Our approach enable synthesizing photo-realistic novel views, while prior methods fail to synthesise high-quality images of detailed scene contents, such as the shirt wrinkles in the first row, the dog’s fur in the second row, and the person’s face and dress in the last row. We refer readers to the supplementary video for full results.

5. Discussion and conclusion

Limitations. Like most prior methods [19, 35], our approach is computationally expensive, is unable to synthesize unseen regions, and is limited to small camera movements compared to representations designed for static scenes. In

Methods	Full			Dynamic Only		
	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓
A) [70]+time	0.905	25.33	0.081	0.683	20.09	0.122
B) w/o TC	0.911	27.57	0.074	0.751	22.16	0.104
C) w/ SF	0.935	29.42	0.035	0.797	22.41	0.095
D) w/o GSE	<u>0.945</u>	<u>30.14</u>	<u>0.032</u>	0.828	24.05	0.070
E) w/o static rep.	0.919	28.19	0.047	0.840	24.01	0.071
F) w/o $\mathcal{L}_{\text{mask}}$	0.930	29.95	0.036	0.835	<u>24.25</u>	0.064
G) w/o \mathcal{L}_{reg}	0.921	29.46	0.042	0.795	22.19	0.080
Full	0.957	30.77	0.028	<u>0.837</u>	24.27	<u>0.066</u>

Table 3. **Ablation study on the Nvidia Dataset.** See Sec. 4.3 for detailed descriptions of each configuration.

addition, compared to prior dynamic NeRF methods, the rendering quality of static scene content depends on which source views are selected. As a result, our method can generate unrealistic background content if insufficient source pixels are aggregated (see supplement for visualization).

Conclusion. We presented a new approach for space-time view synthesis from a monocular video depicting a complex dynamic scene. By representing a dynamic scene within a volumetric IBR framework, our approach overcomes limitations of recent methods that cannot model long videos with complex camera and object motion. We have shown that our method can synthesize photo-realistic novel views from in-the-wild dynamic videos, and can achieve significant improvements over prior state-of-the-art methods on the dynamic scene benchmarks.

References

- [1] Aayush Bansal, Minh Vo, Yaser Sheikh, Deva Ramanan, and Srinivasa Narasimhan. 4D visualization of dynamic events

- from unconstrained multi-view videos. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, pages 5366–5375, 2020.
- [2] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-NeRF 360: Unbounded anti-aliased neural radiance fields. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2022.
 - [3] Mojtaba Bermani, Karol Myszkowski, Hans-Peter Seidel, and Tobias Ritschel. X-fields: Implicit neural view-, light-and time-image interpolation. *ACM Trans. Graphics*, 39(6), 2020.
 - [4] Sai Bi, Zexiang Xu, Pratul P. Srinivasan, Ben Mildenhall, Kalyan Sunkavalli, Milovs Havsan, Yannick Hold-Geoffroy, D. Kriegman, and R. Ramamoorthi. Neural reflectance fields for appearance acquisition. *ArXiv*, abs/2008.03824, 2020.
 - [5] Aljaz Bozic, Michael Zollhofer, Christian Theobalt, and Matthias Niessner. Deepdeform: Learning non-rigid RGB-D reconstruction with semi-supervised data. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, June 2020.
 - [6] 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.
 - [7] Chris Buehler, Michael Bosse, Leonard McMillan, Steven Gortler, and Michael Cohen. Unstructured lumigraph rendering. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 425–432, 2001.
 - [8] Joel Carranza, Christian Theobalt, Marcus A Magnor, and Hans-Peter Seidel. Free-viewpoint video of human actors. *ACM transactions on graphics (TOG)*, 22(3):569–577, 2003.
 - [9] Jin-Xiang Chai, Xin Tong, Shing-Chow Chan, and Heung-Yeung Shum. Plenoptic sampling. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH ’00, page 307–318, USA, 2000. ACM Press/Addison-Wesley Publishing Co.
 - [10] Pierre Charbonnier, Laure Blanc-Feraud, Gilles Aubert, and Michel Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. In *Proceedings of 1st International Conference on Image Processing*, volume 2, pages 168–172. IEEE, 1994.
 - [11] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. MVSNeRF: Fast generalizable radiance field reconstruction from multi-view stereo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14124–14133, 2021.
 - [12] Inchang Choi, Orazio Gallo, Alejandro Troccoli, Min H Kim, and Jan Kautz. Extreme view synthesis. In *Proc. Int. Conf. on Computer Vision (ICCV)*, pages 7781–7790, 2019.
 - [13] Edilson De Aguiar, Carsten Stoll, Christian Theobalt, Naveed Ahmed, Hans-Peter Seidel, and Sebastian Thrun. Performance capture from sparse multi-view video. In *ACM SIGGRAPH 2008 papers*, pages 1–10. 2008.
 - [14] Paul E Debevec, Camillo J Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs: A hybrid geometry-and image-based approach. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 11–20, 1996.
 - [15] Mingsong Dou, Sameh Khamis, Yury Degtyarev, Philip L. Davidson, Sean Ryan Fanello, Adarsh Kowdle, Sergio Orts, Christoph Rhemann, David Kim, Jonathan Taylor, Pushmeet Kohli, Vladimir Tankovich, and Shahram Izadi. Fusion4D: real-time performance capture of challenging scenes. *ACM Trans. Graphics*, 35:114:1–114:13, 2016.
 - [16] Yilun Du, Yanan Zhang, Hong-Xing Yu, Joshua B Tenenbaum, and Jiajun Wu. Neural radiance flow for 4d view synthesis and video processing. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14304–14314. IEEE Computer Society, 2021.
 - [17] John Flynn, Michael Broxton, Paul Debevec, Matthew DuVall, Graham Fyffe, Ryan Overbeck, Noah Snavely, and Richard Tucker. DeepView: View synthesis with learned gradient descent. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, pages 2367–2376, 2019.
 - [18] John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. DeepStereo: Learning to predict new views from the world’s imagery. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, pages 5515–5524, 2016.
 - [19] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. Dynamic view synthesis from dynamic monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5712–5721, 2021.
 - [20] Steven J Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F Cohen. The lumigraph. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 43–54, 1996.
 - [21] Kaiwen Guo, Peter Lincoln, Philip Davidson, Jay Busch, Xueming Yu, Matt Whalen, Geoff Harvey, Sergio Orts-Escolano, Rohit Pandey, Jason Dourgarian, et al. The re-lightables: Volumetric performance capture of humans with realistic relighting. *ACM Transactions on Graphics (ToG)*, 38(6):1–19, 2019.
 - [22] Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. Deep blending for free-viewpoint image-based rendering. *ACM Trans. Graphics*, 37(6):1–15, 2018.
 - [23] Peter Hedman, Tobias Ritschel, George Drettakis, and Gabriel Brostow. Scalable inside-out image-based rendering. *ACM Transactions on Graphics (TOG)*, 35(6):1–11, 2016.
 - [24] Matthias Innmann, Michael Zollhöfer, Matthias Niessner, Christian Theobalt, and Marc Stamminger. VolumeDeform: Real-time volumetric non-rigid reconstruction. In *Proc. European Conf. on Computer Vision (ECCV)*, 2016.
 - [25] Yan-Bin Jia. Plücker coordinates for lines in the space. *Problem Solving Techniques for Applied Computer Science, Com-S-477/577 Course Handout*. Iowa State University. <http://web.cs.iastate.edu/~cs577/handouts/plucker-coordinates.pdf>, 2020.
 - [26] Nima Khademi Kalantari, Ting-Chun Wang, and Ravi Ramamoorthi. Learning-based view synthesis for light field cameras. *ACM Trans. Graphics*, 35(6):1–10, 2016.
 - [27] Takeo Kanade, Peter Rander, and PJ Narayanan. Virtualized reality: Constructing virtual worlds from real scenes. *IEEE multimedia*, 4(1):34–47, 1997.

- [28] Yoni Kasten, Dolev Ofri, Oliver Wang, and Tali Dekel. Layered neural atlases for consistent video editing. *ACM Transactions on Graphics (TOG)*, 40(6):1–12, 2021.
- [29] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.
- [30] Johannes Kopf, Michael F Cohen, and Richard Szeliski. First-person hyper-lapse videos. *ACM Transactions on Graphics (TOG)*, 33(4):1–10, 2014.
- [31] Johannes Kopf, Xuejian Rong, and Jia-Bin Huang. Robust consistent video depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1611–1621, 2021.
- [32] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 31–42, 1996.
- [33] Tianye Li, Mira Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, Richard Newcombe, et al. Neural 3D video synthesis from multi-view video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5521–5531, 2022.
- [34] Zhengqi Li, Tali Dekel, Forrester Cole, Richard Tucker, Noah Snavely, Ce Liu, and William T Freeman. Learning the depths of moving people by watching frozen people. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, pages 4521–4530, 2019.
- [35] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2021.
- [36] Kai-En Lin, Lei Xiao, Feng Liu, Guowei Yang, and Ravi Ramamoorthi. Deep 3D mask volume for view synthesis of dynamic scenes. In *Proc. Int. Conf. on Computer Vision (ICCV)*, 2021.
- [37] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *Advances in Neural Information Processing Systems*, 33:15651–15663, 2020.
- [38] Yuan Liu, Sida Peng, Lingjie Liu, Qianqian Wang, Peng Wang, Christian Theobalt, Xiaowei Zhou, and Wenping Wang. Neural rays for occlusion-aware image-based rendering. In *CVPR*, 2022.
- [39] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *ACM Trans. Graphics*, 38(4):65, 2019.
- [40] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. SMPL: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):1–16, 2015.
- [41] Erika Lu, F. Cole, Tali Dekel, Weidi Xie, Andrew Zisserman, D. Salesin, W. Freeman, and M. Rubinstein. Layered neural rendering for retiming people in video. *ACM Trans. Graphics (SIGGRAPH Asia)*, 2020.
- [42] Erika Lu, Forrester Cole, Tali Dekel, Andrew Zisserman, William T Freeman, and Michael Rubinstein. Omnimate: Associating objects and their effects in video. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2021.
- [43] Xuan Luo, Jia-Bin Huang, Richard Szeliski, Kevin Matzen, and Johannes Kopf. Consistent video depth estimation. *ACM Trans. Graph.*, 39(4), July 2020.
- [44] Ricardo Martin-Brualla, N. Radwan, Mehdi S. M. Sajjadi, J. Barron, A. Dosovitskiy, and Daniel Duckworth. NeRF in the wild: Neural radiance fields for unconstrained photo collections. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2021.
- [45] 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. *Proc. European Conf. on Computer Vision (ECCV)*, 2020.
- [46] Richard A Newcombe, Dieter Fox, and Steven M Seitz. DynamicFusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [47] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3D representations without 3D supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3504–3515, 2020.
- [48] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5865–5874, 2021.
- [49] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M Seitz. HyperNeRF: A higher-dimensional representation for topologically varying neural radiance fields. *arXiv preprint arXiv:2106.13228*, 2021.
- [50] Sida Peng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9054–9063, 2021.
- [51] Eric Penner and Li Zhang. Soft 3D reconstruction for view synthesis. *ACM Trans. Graphics*, 36(6):1–11, 2017.
- [52] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-NeRF: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10318–10327, 2021.
- [53] Gernot Riegler and V. Koltun. Free view synthesis. *Proc. European Conf. on Computer Vision (ECCV)*, 2020.
- [54] Gernot Riegler and Vladlen Koltun. Stable view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12216–12225, 2021.
- [55] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. PIFu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2304–2314, 2019.
- [56] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113, 2016.

- [57] Harry Shum and Sing Bing Kang. Review of image-based rendering techniques. In *Visual Communications and Image Processing 2000*, volume 4067, pages 2–13. SPIE, 2000.
- [58] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33:7462–7473, 2020.
- [59] Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhofer. Deepvoxels: Learning persistent 3D feature embeddings. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, pages 2437–2446, 2019.
- [60] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3D-structure-aware neural scene representations. In *Neural Information Processing Systems*, pages 1119–1130, 2019.
- [61] Pratul P Srinivasan, Richard Tucker, Jonathan T Barron, Ravi Ramamoorthi, Ren Ng, and Noah Snavely. Pushing the boundaries of view extrapolation with multiplane images. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, pages 175–184, 2019.
- [62] Timo Stich, Christian Linz, Georgia Albuquerque, and Marcus Magnor. View and time interpolation in image space. In *Computer Graphics Forum*, volume 27, pages 1781–1787. Wiley Online Library, 2008.
- [63] Mohammed Suhail, Carlos Esteves, Leonid Sigal, and Ameesh Makadia. Light field neural rendering. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2022.
- [64] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems*, 33:7537–7547, 2020.
- [65] Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *Proc. European Conf. on Computer Vision (ECCV)*, 2020.
- [66] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12959–12970, 2021.
- [67] Chaoyang Wang, Ben Eckart, Simon Lucey, and Orazio Gallo. Neural trajectory fields for dynamic novel view synthesis. *arXiv preprint arXiv:2105.05994*, 2021.
- [68] Chaoyang Wang, Xueqian Li, Jhony Kaesemodel Pontes, and Simon Lucey. Neural prior for trajectory estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6532–6542, 2022.
- [69] Liao Wang, Jiakai Zhang, Xinhang Liu, Fuqiang Zhao, Yanshun Zhang, Yingliang Zhang, Minye Wu, Jingyi Yu, and Lan Xu. Fourier plenotrees for dynamic radiance field rendering in real-time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13524–13534, 2022.
- [70] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, pages 4690–4699, 2021.
- [71] Chung-Yi Weng, Brian Curless, Pratul P Srinivasan, Jonathan T Barron, and Ira Kemelmacher-Shlizerman. HumanNeRF: Free-viewpoint rendering of moving people from monocular video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16210–16220, 2022.
- [72] Suttisak Wizadwongsa, Pakkapon Phongthawee, Jiraphon Yenphraphai, and Supasorn Suwajanakorn. NeX: Real-time view synthesis with neural basis expansion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8534–8543, 2021.
- [73] Tianhao Wu, Fangcheng Zhong, Andrea Tagliasacchi, Forrester Cole, and Cengiz Oztireli. D²NeRF: Self-supervised decoupling of dynamic and static objects from a monocular video. *arXiv preprint arXiv:2205.15838*, 2022.
- [74] Wenqi Xian, Jia-Bin Huang, Johannes Kopf, and Changil Kim. Space-time neural irradiance fields for free-viewpoint video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9421–9431, 2021.
- [75] Jae Shin Yoon, Kihwan Kim, Orazio Gallo, Hyun Soo Park, and Jan Kautz. Novel view synthesis of dynamic scenes with globally coherent depths from a monocular camera. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, pages 5336–5345, 2020.
- [76] Jiakai Zhang, Xinhang Liu, Xinyi Ye, Fuqiang Zhao, Yanshun Zhang, Minye Wu, Yingliang Zhang, Lan Xu, and Jingyi Yu. Editable free-viewpoint video using a layered neural representation. *ACM Transactions on Graphics (TOG)*, 40(4):1–18, 2021.
- [77] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, pages 586–595, 2018.
- [78] Zhoutong Zhang, Forrester Cole, Zhengqi Li, Michael Rubinstein, Noah Snavely, and William T Freeman. Structure and motion from casual videos. In *European Conference on Computer Vision*, pages 20–37. Springer, 2022.
- [79] Zhoutong Zhang, Forrester Cole, Zhengqi Li, Noah Snavely, and William Freeman. Structure and motion from casual videos. In *Proc. European Conf. on Computer Vision (ECCV)*, 2022.
- [80] Zhoutong Zhang, Forrester Cole, Richard Tucker, William T Freeman, and Tali Dekel. Consistent depth of moving objects in video. *ACM Transactions on Graphics (TOG)*, 40(4):1–12, 2021.
- [81] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: learning view synthesis using multiplane images. *ACM Trans. Graphics*, 37:1–12, 2018.
- [82] C Lawrence Zitnick, Sing Bing Kang, Matthew Uyttendaele, Simon Winder, and Richard Szeliski. High-quality video

view interpolation using a layered representation. *ACM Trans. Graphics*, 23(3):600–608, 2004.

- [83] Michael Zollhöfer, Matthias Niessner, Shahram Izadi, Christoph Rehmann, Christopher Zach, Matthew Fisher, Chenglei Wu, Andrew Fitzgibbon, Charles Loop, Christian Theobalt, et al. Real-time non-rigid reconstruction using an RGB-D camera. *ACM Trans. Graphics*, 33(4):156, 2014.