Neural Radiance Flow for 4D View Synthesis and Video Processing

Yilun Du
MIT CSAIL

Yinan Zhang
Stanford University

Hong-Xing Yu
Stanford University

Joshua B. Tenenbaum
MIT CSAIL, BCS, CBMM

Jiajun Wu
Stanford University

Abstract

We present a method, Neural Radiance Flow (NeRFlow), to learn a 4D spatial-temporal representation of a dynamic scene from a set of RGB images. Key to our approach is the use of a neural implicit representation that learns to capture the 3D occupancy, radiance, and dynamics of the scene. By enforcing consistency across different modalities, our representation enables multi-view rendering in diverse dynamic scenes, including water pouring, robotic interaction, and real images, outperforming state-of-the-art methods for spatial-temporal view synthesis. Our approach works even when inputs images are captured with only one camera. We further demonstrate that the learned representation can serve as an implicit scene prior, enabling video processing tasks such as image super-resolution and de-noising without any additional supervision [1].

1. Introduction

We live in a rich and dynamic world, consisting of scenes that rapidly change their appearance across both time and view angle. To accurately model the world around us, we need a scene representation that captures underlying lighting, physics, and 3D structure of the scene. Such representations have diverse applications: they can enable interactive exploration in both space and time in virtual reality, the capture of realistic motions for game design, and robot perception and navigation in the environment around them.

Traditional approaches, such as those used in state-of-the-art motion capturing systems, typically are specialized to specific phenomenon [1, 19] and fail to handle complex occlusions and fine details of motion. A core difficulty is that high resolution coverage of information requires a prohibitive amount of memory. Recent work has addressed this by using a neural network as a parametrization for scene details [42, 43, 61]. However, these scene representation often require a large number of images captured from many cameras, which are not generally available in real-world scenarios. Furthermore, most approaches are limited to static scenes, with the exception of X-Fields [43], which have started to consider short-range movements.

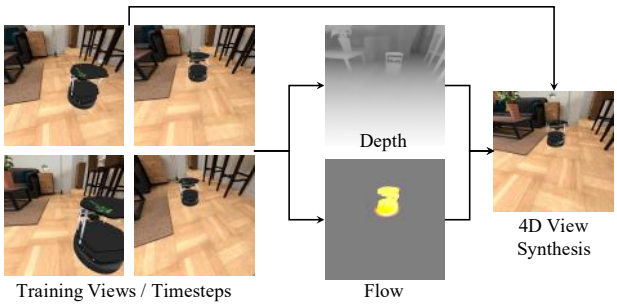


Figure 1: Given a dataset of training images captured from different views and timesteps, NeRFlow learns a spatial-temporal representation that captures the underlying 3D structure and dynamics (with x, y, z directions of flow represented as RGB coordinates respectively) and, in turn, enables 4D view synthesis.

In this work, we aim to learn a dynamic scene representation which allows photorealistic novel view synthesis in complex dynamics, observed by only a limited number of (as few as two) cameras. The key challenge is that the observations at each moment are sparse, restricting prior approaches [42, 61] from fitting a complex scene. To address this problem, we present a novel approach, Neural Radiance Flow (NeRFlow), that can effectively aggregate partial observations across time to learn a coherent spatio-temporal scene representation. We achieve this by formulating a radiance flow field, which encourages temporal consistency of appearance, geometry, and motion.

The radiance flow field is represented by two continuous implicit neural functions: a 6D (spatial position x, y, z, timestep t and viewing direction θ, ϕ) radiance function for

[1] Website at yilundu.github.io/nerflow

geometry and appearance, and a 4D (spatio-temporal position x, y, z, t) flow function for scene dynamics. Our representation enables joint learning of both modules, which is critical given only sparse observations at each moment. Specifically, the flow field provides temporal correspondences for spatial locations, enabling the appearance and geometry information captured at different moments to propagate across time. On the other hand, the radiance function describes the scene geometry that informs the flow module about how the objects in the scene are moving. Our model is fully differentiable, and thus can be trained directly using gradient backpropogation. By learning 3D structure and dynamics, our model can accomplish 4D view synthesis (Figure 1).

To evaluate our approach, we consider three challenging setups: a pouring scene which reflects fluid dynamics, an indoor scene in which a robot walks from near to far to exhibit long-range motion with great occlusion, and multiple complex real scenes with transparent objects. We show that our approach yields high-quality novel view synthesis, even using only two cameras. Our approach outperforms a recent state-of-the-art method [43] on both sufficient and sparse camera settings. In addition, we show that our method can serve as a type of dynamic scene prior, which allows image denoising and super-resolution without any additional supervision, outperforming both classical and state-of-the-art internal learning methods.

In summary, our main contributions are:

1. We present a novel method, Neural Radiance Flow (NeRFlow), for learning implicit spatial-temporal scene representation. It enables novel view synthesis across both space and time, outperforming existing approaches in this area.

2. Our approach can be effective with very limited observations down to only two cameras. We achieve this by introducing a set of temporal consistency constraints over scene appearance, geometry, and motion.

3. We show that our approach can serve as an implicit scene prior, outperforming classical and internal learning methods in super-resolution and image de-noising.

2. Related Works

Neural scene representations. Recently, neural continuous implicit fields [9, 17, 35, 41, 49, 51, 58, 61, 75] have been developed to address the discretization issues and limited resolution of classical 3D representations such as voxel grids [5, 11, 40, 56, 70, 71, 74], point clouds [13, 14, 54, 55] and meshes [18, 26, 28, 34, 67]. Park *et al*. [51] proposed a neural signed distance function to represent scene geometry. Mescheder *et al*. [41] developed neural occupancy fields for scene reconstruction. However, they require groundtruth 3D supervision that can be difficult to obtain.

In order to learn neural scene representations directly from images, differentiable rendering [23, 37, 48, 61] is incorporated to bridge 2D observations and underlying 3D scenes. Sitzmann *et al*. [61] represented scenes with continuous feature fields and propose a neural rendering layer to allow optimization with only posed images. Niemeyer *et al*. [48] used implicit differentiation to bridge 2D images and 3D texture fields. In a recent seminal work, Mildenhall *et al*. [42] introduced a Neural Radiance Field (NeRF) that can be learned using volumetric rendering with only calibrated images. However, these works only consider static scenes.

In contrast, we aim to learn spatial-temporal dynamic scene representations with limited observations. Although it is plausible to extend existing techniques to 4D by assuming a large number of available views at each timestep, we focus on a more realistic setting in capturing dynamic events, where only a few moving cameras are available. Our setup has significance in real-world dynamic event capturing; it also poses a great challenge in aggregating sparse, partial observations across time.

4D reconstruction. Most existing works on spatial-temporal 4D reconstruction either require sufficient observations at each moment [30, 44, 45, 47, 50, 64], or are restricted to specific categories [4, 12, 21, 27, 63, 77] such as human body and faces with template models. Mustafa *et al*. [44, 45] reconstructed dynamic scenes coherently via reconstructing static scenes at each timestep and tracking sparse features to discover correspondences. Niemeyer *et al*. [47] proposed to learn the neural occupancy flow fields using groundtruth 3D supervision. However, these methods need full observations at each timestep, and they do not recover the appearances of the scenes.

Using template models with deformations allows domain knowledge to be easily added and guarantees temporal coherence. Therefore this paradigm is widely adopted for particular shape domains [4, 12, 21, 24, 27, 63, 77] such as human face [4], body [27], and hand [57]. However, these methods depend largely on the quality of template models and it can be costly to obtain high-quality template models beyond the popular shape domains. Unlike these methods, our NeRFlow does not make domain-specific assumptions and is able to learn from limited observations.

Novel view synthesis. Although synthesizing novel views in space [7, 16, 20, 25, 42, 46, 60, 69, 78] or time (*i.e*., video frame interpolation) [3, 22, 39, 62] is widely studied, respectively, spatial-temporal synthesis for dynamic scenes is relatively less explored [32, 79]. Recent works have extended deep learning-based novel view synthesis methods into the temporal domain, by learning a temporal warping function [36, 43] or synthesizing novel views frame-

by-frame [2]. Lombardi *et al*. [36] modeled a scene by a neural feature volume and synthesize novel views at a given moment by sampling the volume with a temporally-specific warping function. Bemana *et al*. [43] targeted at view interpolation across space and time by learning a smart warping function. However, warping-based methods are restricted by input resolution. Our work is different from them in that we learn a continuous implicit representation that can theoretically scale to arbitrary resolution.

Concurrent to our work, several related works [31, 53, 73] also investigate integrating temporal information for sparse time-step novel view synthesis. Core to each approach is the utilization of scene flow, to warp image information across different time-steps. While other approaches rely on discrete warping functions, we learn a continuous warping function that provides continuous key-point correspondences across both space and time. This enables us to generate novel view synthesis across both intermediate viewpoints and time-steps. We further provide additional applications to video processing tasks.

Deep networks as prior. Deep networks have been shown to manifest prior tendency for fitting natural images [33, 65, 66] and temporally-consistent videos [10], even without training on large-scale datasets. This property is referred to as an implicit image/video prior. Similarly, our method can learn neural dynamic scene representations from very sparse observations. To explain such sample efficiency, we posit that our learning method per se may serve as a 'dynamic implicit scene prior'. We validate it by fitting noisy and low-resolution observations, while showing good denoising and super-resolution results. Although our finding shares similar ideas with Ulyanov *et al*. [65], the prior tendency comes from our learning method design instead of network architectures.

3. Neural Radiance Flow (NeRFlow)

Our goal is to learn an implicit neural scene representation for dynamic scenes, even with very limited observations at each moment. A key challenge is to effectively aggregate partial observations from different timesteps while attaining spatio-temporal coherence.

We propose Neural Radiance Flow (NeRFlow) which learns scene appearance, geometry and motion jointly, while encouraging temporal consistency for these components. Specifically, NeRFlow internally represents scene appearance and geometry by a neural radiance field, and it represents scene dynamics by a flow field. These two fields interact to propagate appearance, geometry, and motion information observed at different moments, modulated by a set of consistency losses derived from basic physical intuitions.

We show an overview of our model in Figure 2. In the following, we first describe the radiance field and flow field. Next we present how the overall model is learned jointly using temporal consistency losses. Finally, we outline detail external training supervision from RGB images and overall training implementation.

3.1. Radiance and Flow Fields

NeRFlow consists of two separate modules. The first module, the radiance field, takes a 6D input of position, time, and view direction and outputs emitted color and density, which can be used to form an image by ray marching and volume rendering [42]. The second module, the flow field, takes a 4D input of position and time and outputs its flow or dynamics. We describe each function in detail below.

Radiance field. The radiance function R_θ is a 6-dimensional function, which takes as input the 4D location $\mathbf{x} = (x, y, z, t)$ and 2D viewing direction (θ, ϕ), and outputs an emitted color $\mathbf{c} = (r, g, b)$ and volume density σ, representing the color and transparency of the corresponding 3D point (top of Figure 2). Since geometry is view-independent, we predict the volume density σ independently of view direction. To better aggregate cross-view visual appearance information, we also decompose the predicted color to a view-invariant diffuse part $\mathbf{c}_{\text{diffuse}}$ and a view-dependent specular part $\mathbf{c}_{\text{specular}}$. Since specularity is typically sparsely observed, during training we add an $\mathcal{L}_2$ regularization loss to the magnitude of $\mathbf{c}_{\text{specular}}$.

Flow field. The flow function F_θ represents the underlying dynamics of a scene. F_θ takes as input the 4D location $\mathbf{x} = (x, y, z, t)$. It outputs a flow field

$$\mathbf{f} = (f_x, f_y, f_z) = \left(\frac{\partial x}{\partial t}, \frac{\partial y}{\partial t}, \frac{\partial z}{\partial t}\right), \tag{1}$$

representing instantaneous movement of each point in space. Through integration, this function can then be used to derive the future position of any point. In particular, given a continuous point (x_s, y_s, z_s, t_s), the future position of the point at timestep t_g can be obtained through integration as $(x_s, y_s, z_s) + \int_{t_s}^{t_g} f(x, y, z, t)dt$.

3.2. Temporally Coherent Learning

Throughout learning, we enforce consistency in radiance and flow fields so that they interact to aggregate and propagate partially observed information across time. As shown in Figure 3, this internal learning process is modulated by a set of consistency losses regarding scene appearance, geometry, and motion, following basic physical intuitions. Since we consider sparse observations across times that may not fully cover view angles, we only enforce consistency of diffuse color $\mathbf{c}_{\text{diffuse}}$ (Section 3.1) but not specular color.

Appearance consistency. The diffuse reflectance of an object remains constant while it is moving around. Assuming

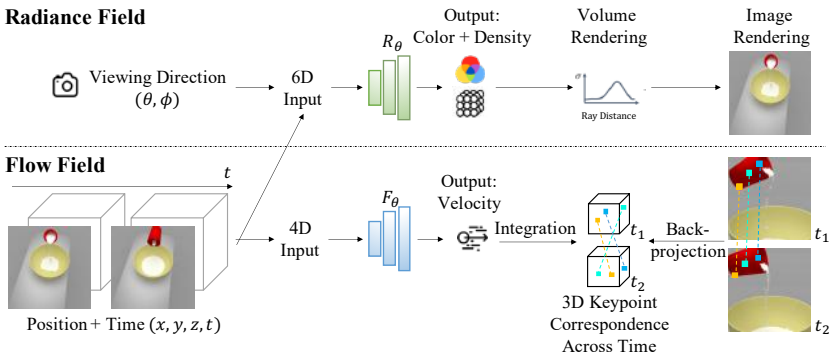


Figure 2: NeRFlow consists of two separate modules, a radiance field (top) trained via neural rendering and a flow field (bottom) trained through 3D key-point correspondence. During testing, we use only the radiance field to synthesize novel images.

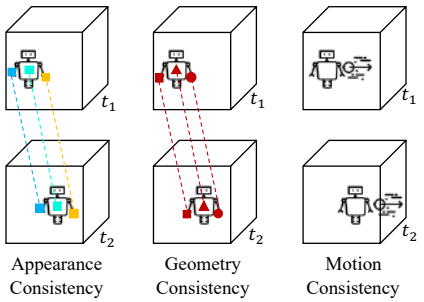


Figure 3: Multiple consistencies are enforced between both radiance and flow fields during training, enabling information radiance information captured at earlier time-steps to inform those of later time-steps.

that the incidental lighting is approximately the same at the object surface, the emitted diffuse color remains constant. Such color constancy assumption is the basis for many optical flow algorithms and approximately holds especially when the motion is small. With this assumption, we develop the appearance consistency loss.

Specifically, we enforce that the diffuse color of points across time are consistent with respect to dynamics predicted by the flow function. In particular, given a randomly sampled 3D point $\mathbf{x}$ at timestep t, we minimize the $\mathcal{L}_2$ distance between the color of $\mathbf{x}$ and that of a computed correspondence $\mathbf{x}_c$ and a future timestep $\mathbf{t}_c$ via

$$\mathcal{L}_{\text{RGB}} = \|\mathbf{c}_{\text{diffuse}}(\mathbf{x}) - \mathbf{c}_{\text{diffuse}}(\mathbf{x}_c)\|, \tag{2}$$

where the flow function F_θ provides point correspondences, given by

$$\mathbf{x}_c = \mathbf{x} + \int_{\mathbf{t}}^{\mathbf{t}_c} F_\theta(x(t))dt. \tag{3}$$

This appearance consistency can be seen as a way of enabling the propagation of color information gathered in earlier (or later) time-step to that of the current time-step. Such propagation of color representations is especially important in settings with limited dynamic cameras where visible frames are entirely disjoint from each other across time.

Geometry consistency. The solidity of an object naturally remains constant while it is moving. Thus, we also enforce that density of points across time are also consistent with respect to dynamics. Analogously to the appearance consistency, we define geometry consistency as

$$\mathcal{L}_{\text{Density}} = \|\sigma(\mathbf{x}) - \sigma(\mathbf{x}_c)\|. \tag{4}$$

We note that the geometry consistency can be particularly useful for particles like fluid, as the shape details in fluid flowing is easily missing without some form of geometry consistency. Our experiment in a pouring scene shows the benefit of our method.

Motion consistency. Our motion consistency is built upon two common physical intuitions: first, empty space looks static, and second, objects move smoothly in natural scenes. For the first assumption, the static empty space should consistently have no motion. Therefore, we enforce that areas with low occupancy must exhibit low flow. To implement this, we cast N query points along a camera ray r, and select the first K query points q_k such that transmittance of the remaining camera points is greater than 0.99. We then penalize the $\mathcal{L}_2$ magnitude of each queried point via $\mathcal{L}_{\text{Flow}} = \|F_\theta(q_k)\|$.

The second assumption can be interpreted as that the overall scenes exhibit relatively low acceleration. Furthermore, a moving object (such as a walking robot) typically manifests similar flow at all points on its surface and within it body. Thus, we encourage the flow function to be smooth across both space and time, by penalizing the gradient of the flow functions at all randomly sampled points $\mathbf{x}$ via $\mathcal{L}_{\text{Acc}} = \|\nabla F_\theta(\mathbf{x})\|^2$.

With these consistency losses, our NeRFlow can learn a spatio-temporally coherent scene representation from limited observations at different moments.

3.3. Learning from Visual Observation

We have introduced the temporally coherent representation of NeRFlow to model dynamic scenes. In the following, we outline the training supervision from visual observations of the scene.

Volume rendering for image supervision. Given posed images (i.e. with camera matrices), we train our model using volumetric rendering following [42]. In particular, let $\{(\mathbf{c}_r^i, \sigma_r^i)\}$, denote the color and volume density of N random samples along a camera ray r. We obtain a RGB value for a pixel through alpha composition across rays:

$$\mathbf{c}_r = \sum_{i=1}^{N} T_r^i \alpha_r^i \mathbf{c}_r^i, \quad T_r^i = \prod_{j=1}^{i-1}(1-\alpha_r^i), \quad \alpha_r^i = 1-\exp(-\sigma_r^i \delta_r^i), \tag{5}$$

Models	Full View				Stereo Views				Dual Views				Sparse Timesteps			
	LPIPS↓	PSNR↑	SSIM↑	MSE↓	LPIPS↓	PSNR↑	SSIM↑	MSE↓	LPIPS↓	PSNR↑	SSIM↑	MSE↓	LPIPS↓	PSNR↑	SSIM↑	MSE↓
Nearest Neighbor	0.1023	25.34	0.9858	0.0051	0.2085	22.89	0.9667	0.0138	0.1305	25.79	0.9789	0.0088	0.1237	24.21	0.9837	0.0061
X-Fields [43]	0.0993	28.83	0.9938	0.0019	0.1261	21.25	0.9809	0.0076	0.1190	20.92	0.9787	0.0082	0.1041	28.65	0.9933	0.0021
NeRFlow w/o Consist.	0.1035	36.30	0.9985	0.0004	0.1219	27.98	0.9942	0.0023	0.1021	31.80	0.9982	0.0006	0.1068	33.75	0.9980	0.0006
NeRFlow (ours)	**0.0980**	**36.57**	**0.9990**	**0.0003**	**0.1170**	**28.29**	**0.9928**	**0.0020**	**0.0851**	**35.29**	**0.9991**	**0.0003**	**0.0949**	**35.87**	**0.9985**	**0.0004**

Table 1: Comparison of our approach with others on the novel-view synthesis setting on the Pouring Dataset.

Models	Full View				Stereo Views				Dual Views				Sparse Timesteps			
	LPIPS↓	PSNR↑	SSIM↑	MSE↓	LPIPS↓	PSNR↑	SSIM↑	MSE↓	LPIPS↓	PSNR↑	SSIM↑	MSE↓	LPIPS↓	PSNR↑	SSIM↑	MSE↓
Nearest Neighbor	0.1945	17.19	0.8728	0.0219	0.3314	15.03	0.8501	0.0422	0.2425	16.86	0.8832	0.0296	0.2084	16.90	0.8698	0.0250
X-Fields [43]	0.2753	21.55	0.9410	0.0096	0.3927	17.75	0.9274	0.0193	0.2587	19.13	0.9370	0.0142	0.2839	21.29	0.9378	0.0106
NeRFlow w/o Consist.	0.1065	29.59	0.9846	**0.0028**	0.2806	22.47	0.9597	0.0070	0.2729	22.26	0.9589	0.0069	0.1130	25.05	0.9712	0.0072
NeRFlow (ours)	**0.0984**	**30.22**	**0.9849**	0.0029	**0.2496**	**23.65**	**0.9690**	**0.0052**	**0.2198**	**24.84**	**0.9758**	**0.0037**	**0.1073**	**25.22**	**0.9717**	**0.0070**

Table 2: Comparison of our approach with others on the novel-view synthesis setting of the Gibson dataset.

where δ_r^i denotes the sampling distance between adjacent points on a ray. We then train our radiance function to minimize the Mean Square Error (MSE) between predicted color and ground truth RGB color via

$$\mathcal{L}_{\text{Render}} = \|\mathbf{c}_r - \text{RGB}\|. \tag{6}$$

Optical flow supervision. In addition to image supervision, we also extract optical flow correspondence using Farnback's method [15] to serve as extra supervision for predicting better scene dynamics. We then obtain sets of 3D spatial-temporal keypoint correspondences using depth maps and camera poses. Given 3D keypoint correspondences between points $\mathbf{x}_s = (x_s, y_s, z_s)$ observed at timestep t_s and $\mathbf{x}_g = (x_g, y_g, z_g)$ observed at timestep t_g, we apply integration using an Runga-Kutta solver on our flow function from point $\mathbf{x}_s$ to obtain a candidate keypoint $\mathbf{x}_g^c$, where $\mathbf{x}_g^c = \mathbf{x}_s + \int_{t_s}^{t_g} F_\theta(x(t))dt$. We then train our flow function to minimize the MSE between predicted and ground truth correspondences via

$$\mathcal{L}_{\text{Corr}} = \|\mathbf{x}_g^c - \mathbf{x}_g\|. \tag{7}$$

3.4. Implementation Details

When training NeRFlow, we first train using $\mathcal{L}_{\text{Render}}$ to warm up training. We then training models with both visual observation losses $\mathcal{L}_{\text{Render}}$ and $\mathcal{L}_{\text{Corr}}$ and consistency losses $\mathcal{L}_{\text{RGB}}$, $\mathcal{L}_{\text{Density}}$, $\mathcal{L}_{\text{Flow}}$ and $\mathcal{L}_{\text{Acc}}$ for an overall training loss of $\mathcal{L}_{\text{Render}} + \mathcal{L}_{\text{Corr}} + \mathcal{L}_{\text{Density}} + \mathcal{L}_{\text{Flow}} + \mathcal{L}_{\text{Acc}}$.

We utilize the trigonometric frequency embedding in NeRF [42] in radiance functions to enable the radiance function to capture high resolution details in a scene. We omit the frequency embedding in flow functions to improve the smoothness of flow predictions in NeRFlow.

We parameterize both radiance and flow functions using 8 layer MLPs with 256 hidden dimensions. To accomplish decomposition of color, the radiance function F_θ first processes the input 4D coordinates $\mathbf{x}$ with 8 fully-connected layers and predicts σ, $\mathbf{c}_{\text{diffuse}}$ and a 256 dimensional hidden vector. This hidden vector is then concatenated with the view vector to predict $\mathbf{c}_{\text{specular}}$.

4. Experiments

We validate the performance of NeRFlow on representing dynamic scenes of pouring [59], iGibson [72], and real images from [43] and [38] through multi-view rendering. We further show that our approach infers high quality depth and flow maps. Finally, we show that NeRFlow can serve as a scene prior, denoising noisy images and enabling image super-resolution.

4.1. 4D View Synthesis

Data. We validate our approach on three different sets of dynamic scenes.

1. **Pouring**: The pouring scene contains fluid dynamics [59]. We render images at 400×400 pixels. We utilize a training set size of 1,000 images (unless otherwise noted), and test set of 100 images.

2. **Gibson**: The Gibson scene has a robot walking a long distance. We render images on the iGibson environment [72], using the Rs_interactive scene with a robot TurtleBot moving linearly on the floor. Each image is rendered at 800×800 pixels. We use a training set size of 300 images and test set size of 100 images.

3. **Real Images**: We utilize two real dynamic scenes from [43], including a complex indoor scene and one with transparent objects. We split 90% of provided images as the training set and the remaining 10% as the test set, and use Colmap to obtain poses for images. We further qualitatively evaluate our approach on monocular real world video from [38].

Metrics and baselines. To measure the performance of our approach, we report novel view synthesis performance

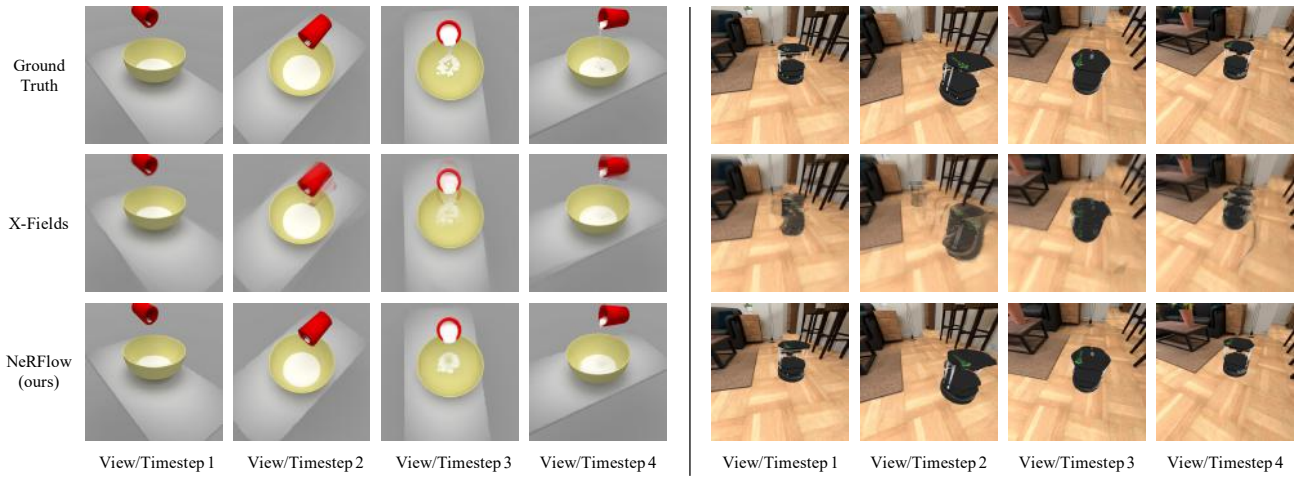
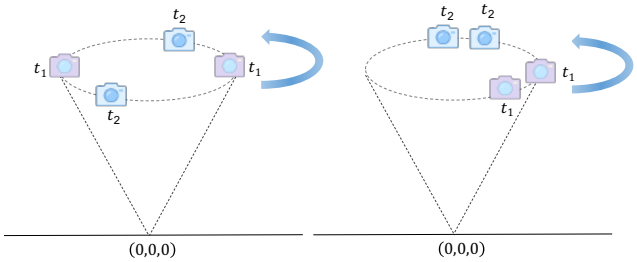


Figure 4: Comparison of our approach with others on Pouring and Gibson in the 'Full View' setting.



(a) Dual Views (b) Stereo Views

Figure 5: Illustration of cameras in the limited views setting.

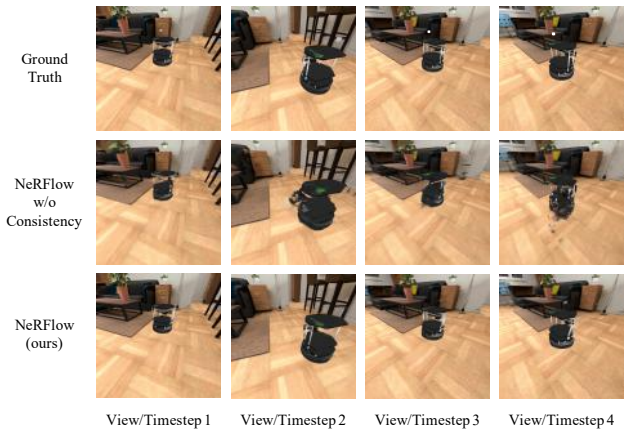


Figure 6: Illustration of renderings from our model with or without consistency regularization in Gibson setting, where the scene is captured by two stereo cameras. Consistency regularization enables renderings to reasonable in extrapolated viewpoints.

using LPIPS [76], PSNR, SSIM [68], and MSE. We compare our multi-view rendering results against the nearest neighbor baseline from Open4D [2] (using VGG feature distance), and against a recent state-of-the-art method, X-Fields [43], which relies on warping existing training images to synthesize novel views.

Results on full view synthesis. We first investigate novel view synthesis in the setting where poses of multi-view training images are drawn uniformly across time. For the Pouring dataset, we sample cameras poses randomly in the upper hemisphere. For the Gibson dataset, we sample cameras from a set of forward facing scenes.

Tables 1 and 2 ('Full View') include quantitative results on the Pouring and Gibson datasets, respectively. Figure 4 shows quantitative results on both datasets. We find that NeRFlow outperforms baselines in all metrics. On Pouring, where modeling fluid dynamics is difficult, NeRFlow is able to capture the fluid splatter pattern and dynamics (Figure 4 left). On Gibson, which exhibits long range motion and occlusion (Figure 4 right), NeRFlow is able to handle of occlusions of robot. Finally on real images, we find NeRFlow is exhibits significantly less ghosting than baselines.

Results in limited camera view settings. We next consider novel view synthesis in the setting where multi-view training images are captured by two moving cameras. We consider two different camera setups. In the stereo setting, training images are captured by two nearby cameras that are rotating together around a circle over time, while in the dual setting, training images are captured by two diametrically opposite cameras that are rotating together around a circle over time. We illustrate both camera settings in Figure 5. We test novel synthesis from random views taken on any location of the circle across any time. To accomplish this task well, a model must learn to integrate the radiance information captured across different timesteps together.

We report the results in Tables 1 and 2 ('Stereo Views' and 'Opposite Views'). In this setting, we find that consistency enables our approach to do significantly better quantitatively. We show qualitative rendering results in this setting in Figure 6. Consistency enables more effective propagation of

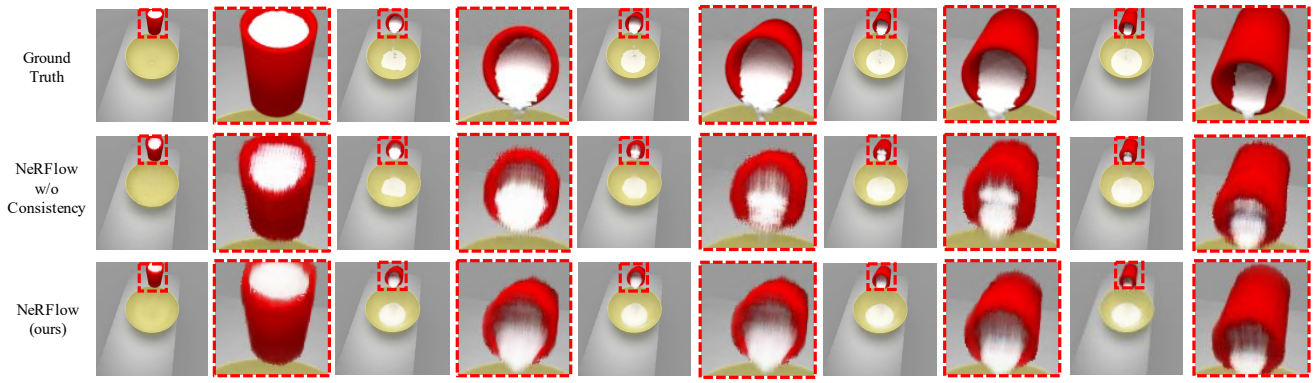


Figure 7: Illustration of rendering results on intermediate timesteps when models are trained with a sparse number of timesteps. Consistency makes pouring volume significantly more stable.

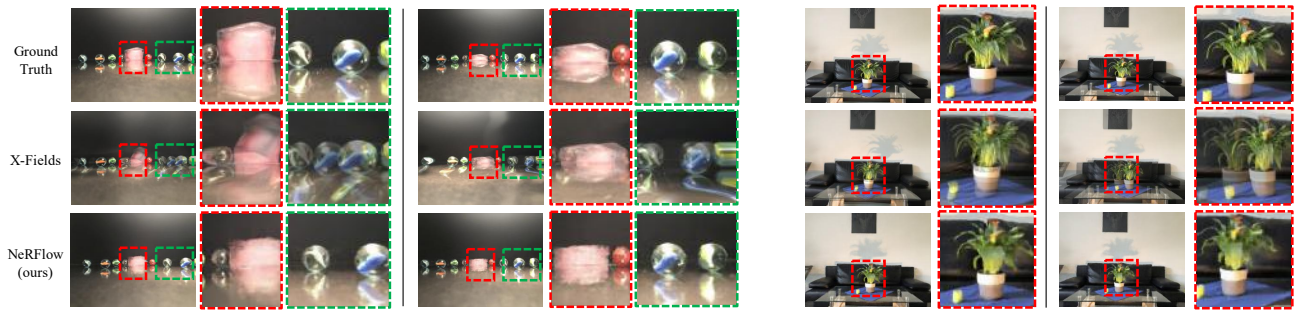


Figure 8: Comparison of our NeRFlow with X-Fields [43] on real images that include ice (left) and a vase (right).

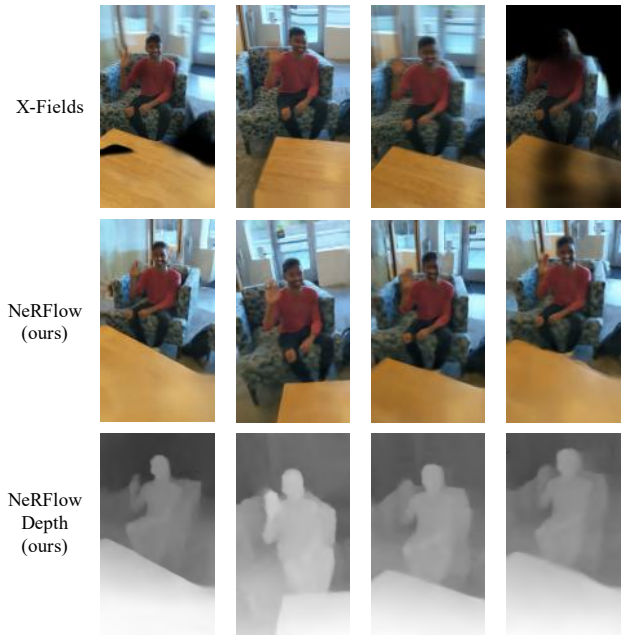


Figure 9: Novel view synthesis and estimated depth maps on a monocular video from [38] of NeRFlow compared to XFields.

information across time. It makes renderings exhibit more consistent fluid placement on Pouring, and produces sharp renderings of the moving robot on Gibson.

Results on temporal interpolation. We further consider 4D view synthesis, where multi-view training images are drawn from a fixed, sparse subset of all timesteps in the scene. In particular, on Pouring, we train models with 1 of every 10 time steps; on Gibson, we draw 1 out of every 5 timesteps during training. We measure test performance in rendering at arbitrary timesteps in our scene. Such a task tests the temporal interpolation capability of our model, a useful ability for applications such slow motion generation and frame rate up-conversion.

We report quantitative results in Tables 1 and 2 ('Sparse Timesteps') and find that consistency boosts the rendering performance. Consistency constrains radiance fields to change smoothly with respect to time, enabling smooth renderings of intermediate timesteps. Qualitatively, we illustrate this effect in Figure 7. We find that in synthesized views from NeRFlow without consistency, fluid levels fluctuate through time, while in synthesized views from NeRFlow with consistency, fluid levels decrease smoothly.

Results on real images. We further validate our approach on real images, where input views are sparse. We utilize the dataset provided in [43] and provide results in Table 3. Qualitatively, we visualize our outputs in Figure 8. In the real image setting, we find that our approach can capture transparency and various lighting effect while X-Fields strug-

Dataset	Model	LPIPS↓	PSNR↑	SSIM↑	MSE↓
Ice	X-Fields [43]	0.2271	18.69	0.9347	0.0140
	NeRFlow (Ours)	**0.2031**	**29.04**	**0.9922**	**0.0012**
Vase	X-Fields [43]	0.2151	19.92	0.9259	0.0105
	NeRFlow (Ours)	**0.1972**	**28.91**	**0.9851**	**0.0013**

Table 3: Comparison of our approach with X-Fields [43] on 4D view synthesis on real images.

Models	Full View	Stereo Views	Opposite Views
NeRFlow w/o Consistency	0.3747	0.4433	0.4003
NeRFlow (ours)	**0.3692**	**0.2701**	**0.2675**

Table 4: Evaluation of depth estimation of NeRFlow with or without physical constraints. We report MSE error with ground truth depth.

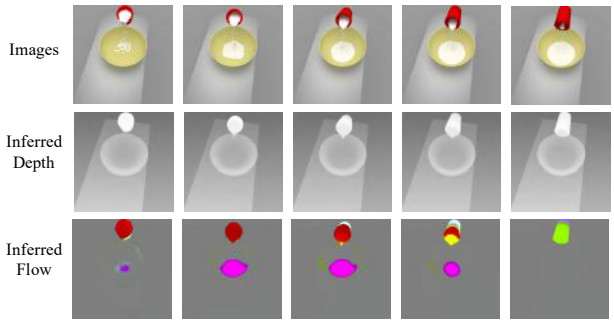


Figure 10: Examples of estimated depth and flow (with x, y, z directions of flow represented as RGB coordinates respectively) on the Pouring scenes.

gles with ghosting. We further qualitatively evaluate our approach on monocular real video from [38] provide results in Figure 9.

4.2. Scene Structure and Dynamics Estimation

We next examine the ability of NeRFlow to capture the underlying 3D structure and dynamics of a scene. We quantitatively measure the depth estimation accuracy of NeRFlow in Table 4 in terms of MSE with respect to ground truth depth. By enforcing geometric constancy across time, we find that our consistency loss improves depth estimates from NeRFlow, especially in limited camera settings. In Figure 10, we visualize inferred depth and flow fields over time in pouring. We find that the inferred flow field captures the dynamics of pouring, including the flow of liquid as well as the movement of the cup. Similarly, we visualized predicted depth on real video in Figure 9. Similarly we find that inferred depth appears to match the relative ground truth of the scene.

4.3. Video Processing

Given a set of images capturing a dynamic scene, NeRFlow learns to represent the underlying 3D structure and its evolution through time. This scene description can be seen as a scene prior, where by utilizing volumetric rendering on our scene description, we accomplish additional video processing tasks such as video denoising and super-resolution.

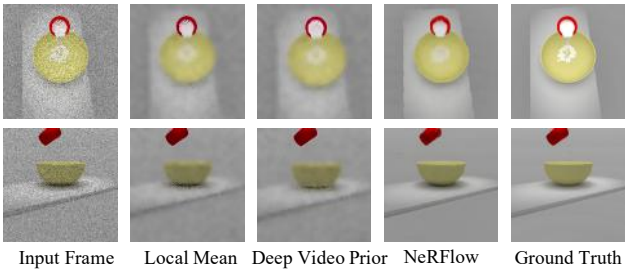


Figure 11: Examples of video denoising using our approach compared to other approaches.

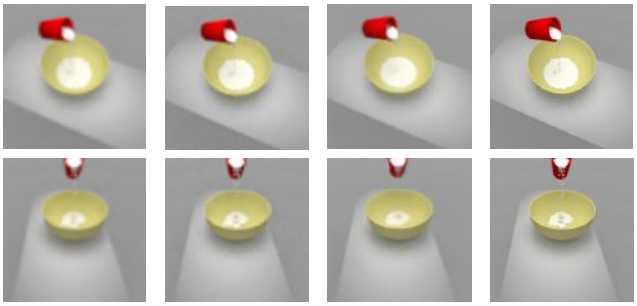
Models	LPIPS↓	PSNR↑	SSIM↑	MSE↓
Non-Local Means [6]	0.3051	23.49	0.9856	0.0046
Blind Video Prior [10]	0.2707	21.67	0.9797	0.0070
NeRFlow (ours)	**0.1372**	**27.71**	**0.9949**	**0.0018**

Table 5: Results of NeRFlow, Blind Video Prior [10], and Non-Local Means [6] on the task of denoising.

Datasets. We evaluate our approach on the tasks of video de-noising and image super-resolution. To test de-noising, we train our model on 1,000 pouring images of the same scene with a resolution of 400×400, rendered with a 2 ray-casts (compared with 128 rays used in Section 4.1) in Blender, and test the difference between the rendered images and the ground truth images obtained from Blender using 128 ray-casts. To test super-resolution, we train our model on 1,000 pouring images of the same scenewith a resolution of 64×64 and test rendering of images of size 200×200. Finally, we evaluate our approach on de-noising a monocular real video (Ayush) from [38], where we corrupt input frames with Gaussian noise with standard deviation of 25.

Baselines. We compare with the very recent, state-of-the-art internal learning method, Blind Video Prior [10], which uses a learned network to approximate a task mapping. During training, we supervise the Blind Video Prior on denoising and super-resolution using the outputs of the classical algorithms: Non-Local Means [6] for denoising and bi-cubic interpolation for super-resolution. We also compare with these classical algorithms directly.

Video denoising. We quantitatively compare our approach with the baselines on image de-noising in Table 5. We find that our approach achieves more realistic images than the baselines, as well as lower reconstruction error. We illustrate qualitative examples of our generation in Figure 11 and find that our approach is able to remove most of the noise in the input images. By accumulating radiance information across different input images, our representation learns to remove the overall image noise.



Input Frame | Deep Video Prior | NeRFlow | Ground Truth

Figure 12: Examples of super-resolution using our approach compared to other approaches using internal learning.

Model	LPIPS↓	PSNR↑	SSIM↑	MSE↓
Bicubic Interpolation	0.1427	30.27	0.9961	0.0012
Blind Video Prior [10]	0.1870	30.58	**0.9963**	**0.0009**
NeRFlow (ours)	**0.0903**	**30.67**	**0.9963**	**0.0009**

Table 6: Results of NeRFlow, Blind Video Prior [10], and bi-cubic interpolation on the task of image super-resolution.

Models	LPIPS↓	PSNR↑	SSIM↑	MSE↓
Non-Local Means [6]	0.4662	24.91	0.9263	0.0032
Blind Video Prior [10]	0.5572	18.24	0.8891	0.0151
NeRFlow (ours)	**0.3556**	**28.46**	**0.9837**	**0.0014**

Table 7: Video denoising results of NeRFlow, Blind Video Prior [10], and Non-Local Means [6] on the real monocular video from [38].

Real monocular video. We next evaluate our approach on image de-noising of Gaussian noise applied to the real monocular video in Table 7. We find that our approach is able to obtain more realistic images than our baselines (as determined by LPIPS) and achieves lower MSE.

Video super-resolution. We finally evaluate our approach on image super-resolution with our baselines in Table 6. We find that in this setting our approach again achieves more realistic images than our baselines (as determined by LPIPS), with the Blind Video Prior achieving comparable image MSE. When rendering higher resolution images from our radiance function, representations in NeRFlow have accumulated radiance information across different input images, and are capable of rendering higher resolution details, despite being only trained on low resolution images. In Figure 12, we illustrate renderings of our model and other baselines. Compared to our baselines, NeRFlow exhibits less noise in the background, and captures sharper details along the pouring bowl.

5. Conclusion

We have presented NeRFlow, a method that learns a powerful spatial-temporal representation of a dynamic scene. We show that NeRFlow can be used for 4D view synthesis in the setting of limited cameras on three datasets, and that it can further capture both underlying depth and dynamics information in a scene. Finally, we show that NeRFlow can serve as a learned scene prior, which can be applied to video processing tasks such as video de-noising and super-resolution.

References

[1] Bradley Atcheson, Ivo Ihrke, Wolfgang Heidrich, Art Tevs, Derek Bradley, Marcus Magnor, and Hans-Peter Seidel. Time-resolved 3d capture of non-stationary gas flows. *ACM transactions on graphics (TOG)*, 27(5):1–9, 2008. 1

[2] Aayush Bansal, Minh Vo, Yaser Sheikh, Deva Ramanan, and Srinivasa Narasimhan. 4d visualization of dynamic events from unconstrained multi-view videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5366–5375, 2020. 3, 6

[3] Wenbo Bao, Wei-Sheng Lai, Chao Ma, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Depth-aware video frame interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3703–3712, 2019. 2

[4] Amit Bermano, Thabo Beeler, Yeara Kozlov, Derek Bradley, Bernd Bickel, and Markus Gross. Detailed spatio-temporal reconstruction of eyelids. *ACM Transactions on Graphics (TOG)*, 34(4):1–11, 2015. 2

[5] Andrew Brock, Theodore Lim, James M Ritchie, and Nick Weston. Generative and discriminative voxel modeling with convolutional neural networks. *arXiv preprint arXiv:1608.04236*, 2016. 2

[6] Antoni Buades, Bartomeu Coll, and J-M. Morel. A non-local algorithm for image denoising. In *2005 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005. 8, 9

[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. 2

[8] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *Advances in neural information processing systems*, pages 6571–6583, 2018. 13

[9] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5939–5948, 2019. 2

[10] Qifeng Chen Chenyang Lei, Yazhou Xing. Blind video temporal consistency via deep video prior. *https://arxiv.org/pdf/2010.11838.pdf*, 2020. 3, 8, 9

[11] 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 *ECCV*, 2016. 2

[12] Huseyin Coskun, Felix Achilles, Robert DiPietro, Nassir Navab, and Federico Tombari. Long short-term memory kalman filters: Recurrent neural estimators for pose regularization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5524–5532, 2017. 2

[13] Gil Elbaz, Tamar Avraham, and Anath Fischer. 3d point cloud registration for localization using a deep neural network auto-encoder. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4631–4640, 2017. 2

[14] Haoqiang Fan, Hao Su, and Leonidas Guibas. A point set generation network for 3d object reconstruction from a single image. In *CVPR*, 2017. 2

[15] Gunnar Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Scandinavian conference on Image analysis*, pages 363–370. Springer, 2003. 5

[16] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2367–2376, 2019. 2

[17] Kyle Genova, Forrester Cole, Daniel Vlasic, Aaron Sarna, William T Freeman, and Thomas Funkhouser. Learning shape templates with structured implicit functions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7154–7164, 2019. 2

[18] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. A papier-mâché approach to learning 3d surface generation. In *CVPR*, 2018. 2

[19] Tim Hawkins, Per Einarsson, and Paul Debevec. Acquisition of time-varying participating media. *ACM Transactions on Graphics (ToG)*, 24(3):812–815, 2005. 1

[20] Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. Deep blending for free-viewpoint image-based rendering. *ACM Transactions on Graphics (TOG)*, 37(6):1–15, 2018. 2

[21] Yinghao Huang, Federica Bogo, Christoph Lassner, Angjoo Kanazawa, Peter V Gehler, Javier Romero, Ijaz Akhter, and Michael J Black. Towards accurate marker-less human shape and pose estimation over time. In *2017 international conference on 3D vision (3DV)*, pages 421–430. IEEE, 2017. 2

[22] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. Super slomo: High quality estimation of multiple intermediate frames for video interpolation, 2018. 2

[23] Yue Jiang, Dantong Ji, Zhizhong Han, and Matthias Zwicker. Sdfdiff: Differentiable rendering of signed distance fields for 3d shape optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1251–1261, 2020. 2

[24] Hanbyul Joo, Tomas Simon, and Yaser Sheikh. Total capture: A 3d deformation model for tracking faces, hands, and bodies. In *CVPR*, 2018. 2

[25] Nima Khademi Kalantari, Ting-Chun Wang, and Ravi Ramamoorthi. Learning-based view synthesis for light field cameras. *ACM Transactions on Graphics (TOG)*, 35(6):1–10, 2016. 2

[26] Angjoo Kanazawa, Shubham Tulsiani, Alexei A Efros, and Jitendra Malik. Learning category-specific mesh reconstruction from image collections. *arXiv:1803.07549*, 2018. 2

[27] Angjoo Kanazawa, Jason Y Zhang, Panna Felsen, and Jitendra Malik. Learning 3d human dynamics from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5614–5623, 2019. 2

[28] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In *CVPR*, 2018. 2

[29] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 13

[30] Vincent Leroy, Jean-Sébastien Franco, and Edmond Boyer. Multi-view dynamic shape refinement using local temporal integration. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3094–3103, 2017. 2

[31] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. *arXiv preprint arXiv:2011.13084*, 2020. 3

[32] Christian Lipski, Christian Linz, Kai Berger, Anita Sellent, and Marcus Magnor. Virtual video camera: Image-based viewpoint navigation through space and time. In *Computer Graphics Forum*, 2010. 2

[33] Jiaming Liu, Yu Sun, Xiaojian Xu, and Ulugbek S Kamilov. Image restoration using total variation regularized deep image prior. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7715–7719. IEEE, 2019. 3

[34] Shichen Liu, Weikai Chen, Tianye Li, and Hao Li. Soft rasterizer: Differentiable rendering for unsupervised single-view mesh reconstruction. *arXiv preprint arXiv:1901.05567*, 2019. 2

[35] Shichen Liu, Shunsuke Saito, Weikai Chen, and Hao Li. Learning to infer implicit surfaces without 3d supervision. In *Advances in Neural Information Processing Systems*, pages 8295–8306, 2019. 2

[36] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *arXiv preprint arXiv:1906.07751*, 2019. 2, 3

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

[38] Xuan Luo, Jia-Bin Huang, Richard Szeliski, Kevin Matzen, and Johannes Kopf. Consistent video depth estimation. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH)*, 39(4), 2020. 5, 7, 8, 9, 13

[39] Dhruv Mahajan, Fu-Chung Huang, Wojciech Matusik, Ravi Ramamoorthi, and Peter Belhumeur. Moving gradients: a path-based method for plausible image interpolation. *ACM Transactions on Graphics (TOG)*, 28(3):1–11, 2009. 2

[40] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *IROS*, 2015. 2

[41] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019. 2

[42] 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, 2, 3, 4, 5

[43] Bemana Mojtaba, Myszkowski Karol, Seidel Hans-Peter, and Ritschel Tobias. X-fields: Implicit neural view-, light- and

time-image interpolation. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia 2020)*, 39(6), 2020. 1, 2, 3, 5, 6, 7, 8

[44] Armin Mustafa, Hansung Kim, Jean-Yves Guillemaut, and Adrian Hilton. General dynamic scene reconstruction from multiple view video. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 900–908, 2015. 2

[45] Armin Mustafa, Hansung Kim, Jean-Yves Guillemaut, and Adrian Hilton. Temporally coherent 4d reconstruction of complex dynamic scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4660–4669, 2016. 2

[46] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. Hologan: Unsupervised learning of 3d representations from natural images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7588–7597, 2019. 2

[47] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Occupancy flow: 4d reconstruction by learning particle dynamics. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5379–5389, 2019. 2

[48] 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. 2

[49] Michael Oechsle, Lars Mescheder, Michael Niemeyer, Thilo Strauss, and Andreas Geiger. Texture fields: Learning texture representations in function space. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4531–4540, 2019. 2

[50] Martin Ralf Oswald, Jan Stühmer, and Daniel Cremers. Generalized connectivity constraints for spatio-temporal 3d reconstruction. In *European Conference on Computer Vision*, pages 32–46. Springer, 2014. 2

[51] 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. 2

[52] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 13

[53] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. *arXiv preprint arXiv:2011.13961*, 2020. 3

[54] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NIPS*, 2017. 2

[55] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108, 2017. 2

[56] Gernot Riegler, Ali Osman Ulusoys, and Andreas Geiger. Octnet: Learning deep 3d representations at high resolutions. In *CVPR*, 2017. 2

[57] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics (ToG)*, 36(6):245, 2017. 2

[58] 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 International Conference on Computer Vision*, pages 2304–2314, 2019. 2

[59] Connor Schenck and Dieter Fox. Towards learning to perceive and reason about liquids. In *International Symposium on Experimental Robotics*, pages 488–501. Springer, 2016. 5

[60] Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhofer. Deepvoxels: Learning persistent 3d feature embeddings. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2437–2446, 2019. 2

[61] 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, 2

[62] Shao-Hua Sun, Minyoung Huh, Yuan-Hong Liao, Ning Zhang, and Joseph J Lim. Multi-view to novel view: Synthesizing novel views with self-learned confidence. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 155–171, 2018. 2

[63] Hsiao-Yu Tung, Hsiao-Wei Tung, Ersin Yumer, and Katerina Fragkiadaki. Self-supervised learning of motion capture. In *NIPS*, 2017. 2

[64] Ali Osman Ulusoy, Octavian Biris, and Joseph L Mundy. Dynamic probabilistic volumetric models. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 505–512, 2013. 2

[65] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9446–9454, 2018. 3

[66] Dave Van Veen, Ajil Jalal, Mahdi Soltanolkotabi, Eric Price, Sriram Vishwanath, and Alexandros G Dimakis. Compressed sensing with deep image prior and learned regularization. *arXiv preprint arXiv:1806.06438*, 2018. 3

[67] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. *arXiv:1804.01654*, 2018. 2

[68] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 13(4):600–612, 2004. 6

[69] 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. 2

[70] Jiajun Wu, Chengkai Zhang, Tianfan Xue, William T Freeman, and Joshua B Tenenbaum. Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling. In *NIPS*, 2016. 2

[71] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *CVPR*, 2015. 2

[72] Fei Xia, William B Shen, Chengshu Li, Priya Kasimbeg, Micael Edmond Tchapmi, Alexander Toshev, Roberto Martín-Martín, and Silvio Savarese. Interactive gibson benchmark: A benchmark for interactive navigation in cluttered environments. *IEEE Robotics and Automation Letters*, 5(2):713–720, 2020. 5

[73] Wenqi Xian, Jia-Bin Huang, Johannes Kopf, and Changil Kim. Space-time neural irradiance fields for free-viewpoint video. *arXiv preprint arXiv:2011.12950*, 2020. 3

[74] Haozhe Xie, Hongxun Yao, Xiaoshuai Sun, Shangchen Zhou, and Shengping Zhang. Pix2vox: Context-aware 3d reconstruction from single and multi-view images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2690–2698, 2019. 2

[75] Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. In *Advances in Neural Information Processing Systems*, pages 492–502, 2019. 2

[76] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep networks as a perceptual metric. In *CVPR*, 2018. 6

[77] Qian Zheng, Xiaochen Fan, Minglun Gong, Andrei Sharf, Oliver Deussen, and Hui Huang. 4d reconstruction of blooming flowers. In *Computer Graphics Forum*, 2017. 2

[78] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A Efros. View synthesis by appearance flow. In *ECCV*, 2016. 2

[79] C Lawrence Zitnick, Sing Bing Kang, Matthew Uyttendaele, Simon Winder, and Richard Szeliski. High-quality video view interpolation using a layered representation. *ACM TOG*, 23(3):600–608, 2004. 2

A.1. Appendix

A.1.1. Additional Experimental Details

We implement and train models using the Pytorch framework [52]. Models are trained for 10 hours with only $\mathcal{L}_{\text{Render}}$, and then trained subsequently for another 10 hours with all losses. Models on trained on single GPU. Models are trained with a learning rate of 1e-3, with a exponential learning rate decay by a factor 0.1 every 40000 steps using the Adam optimizer [29].

To integrate flow for temporal correspondences, we utilize the Neural ODE library [8], with the Runga-Kutta solver, with a rtol of $1e-4$ and atol of $1e-5$. While these values are higher than typical values used during Neural ODE training, we found that they were critical towards stable flow inference. Smaller values cause much slower training times and lead to less smooth flow fields. We rescale timesteps to be between -1 and 1. When enforcing consistency through $\mathcal{L}_{\text{Corr}}$ and $\mathcal{L}_{\text{Density}}$, initial points for determining correspondence are sampled from the surface of the 4D scene (as determined by volumetric occupancy when casting rays), with correspondence enforced between temporal differences randomly drawn from 0.0 and 0.4.

When training models, we penalize the predictions of $\mathbf{c}_{\text{specular}}$ using a L2 coefficient of 0.1. We apply coefficients of 0.01 for $\mathcal{L}_{\text{Flow}}$, 0.1 for $\mathcal{L}_{\text{Acc}}$, and 0.001 for $\mathcal{L}_{\text{Corr}}$ and $\mathcal{L}_{\text{Density}}$. In real world settings without depth images, we determine 3D correspondences by backprojecting each pixel in optical flow to its estimated depth under the NeRF model. To enforce that the estimated depth is reasonable, we add an additional L2 loss term enforcing whitened (normalized to mean 0 and standard deviation 1) predicted depth matches whitened estimated depth from [38].