Non-Rigid Neural Radiance Fields: Reconstruction and Novel View Synthesis of a Dynamic Scene From Monocular Video

EDGAR TRETSCHK, AYUSH TEWARI, and VLADISLAV GOLYANIK, MPI for Informatics, SIC
MICHAEL ZOLLHÖFER and CHRISTOPH LASSNER, Facebook Reality Labs Research
CHRISTIAN THEOBALT, MPI for Informatics, SIC

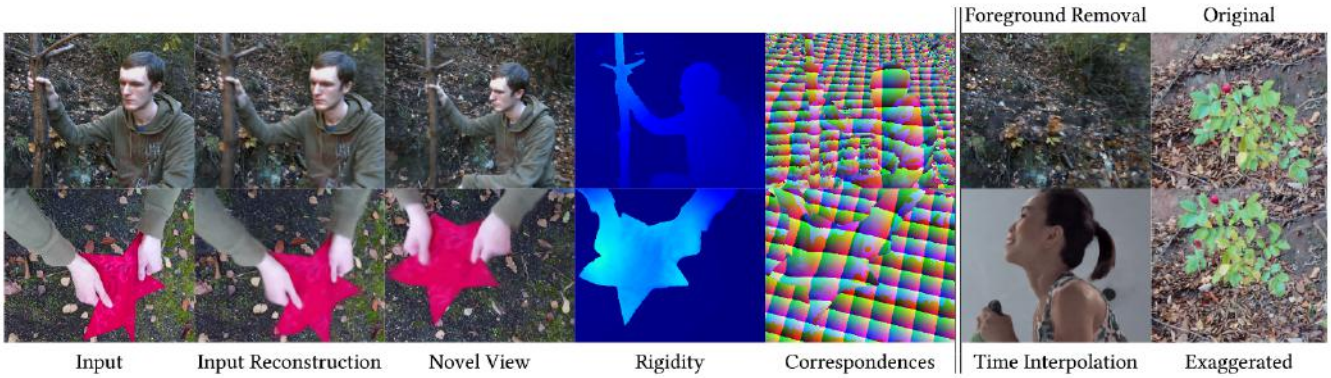


Fig. 1. Given a monocular image sequence, our approach reconstructs a single canonical neural radiance field to represent geometry and appearance, and a per-time-step deformation field of the entire scene, thereby decoupling the deformations from geometry and appearance. The deformation is realized as a coordinate-based MLP that takes points from the deformed scene state to the canonical neural radiance field. Starting with point samples from a straight ray in the deformed scene state for volumetric rendering, this MLP effectively bends the ray into the canonical model. This representation enables input reconstruction and novel view synthesis with significant differences between input camera views and novel views. In addition, our learned representation provides rigidity scores and correspondences, without any direct supervision on either. We can further use the rigidity scores to remove the foreground, we can supersample along the temporal dimension, and we can exaggerate or dampen motion.

We present Non-Rigid Neural Radiance Fields (NR-NeRF), a reconstruction and novel view synthesis approach for general non-rigid dynamic scenes. Our approach takes RGB images of a dynamic scene as input, e.g., from a monocular video recording, and creates a high-quality space-time geometry and appearance representation. In particular, we show that even a single handheld consumer-grade camera is sufficient to synthesize sophisticated renderings of a dynamic scene from novel virtual camera views, for example a 'bullet-time' video effect. Our method disentangles the dynamic scene into a canonical volume and its deformation. Scene deformation is implemented as ray bending, where straight rays are deformed non-rigidly to represent scene motion. We also propose a novel rigidity regression network that enables us to better constrain rigid regions of the scene, which leads to more stable results. The ray bending and rigidity network are trained without any explicit supervision. In addition to novel view synthesis, our formulation enables dense correspondence estimation across views and time, as well as compelling video editing applications such as motion exaggeration. We demonstrate the effectiveness of our method using extensive evaluations, including ablation studies and comparisons to the state of the art. We urge the reader to watch the supplemental video for qualitative results. Our code will be open sourced.

ACM Reference Format:
Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. 2021. Non-Rigid Neural Radiance Fields: Reconstruction and Novel View Synthesis of a Dynamic Scene From Monocular Video. 1, 1 (March 2021), 17 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

2021. XXXX-XXXX/2021/3-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Motivation. Free viewpoint rendering is a well-studied problem due to its wide range of applications in movies and virtual/augmented reality [Collet et al. 2015; Miller et al. 2005; Smolic et al. 2006]. In this work, we are interested in dynamic scenes, which change over time, from novel user-controlled viewpoints. Traditionally, multi-view recordings are required for free viewpoint rendering of dynamic scenes [Oswald et al. 2014; Tung et al. 2009; Zhang et al. 2003]. However, such multi-view captures are expensive and cumbersome, and thus not suitable for casual users. We would like to enable the setting in which a user records a dynamic scene with a single, moving consumer-grade camera. Free viewpoint rendering allows for a more immersive experience than 2D video, and would even allow for an immersive experience of existing videos recorded in the distant past.

Access to only a monocular video of the deforming scene leads to a severely under-constrained problem. Most existing approaches thus limit themselves to a single object category, such as the human body [Habermann et al. 2019; Kocabas et al. 2020; Xiang et al. 2019] or face [Egger et al. 2020]. Some approaches allow for the reconstruction of general non-rigid objects [Garg et al. 2013; Kumar et al. 2018; Sidhu et al. 2020; Zollhöfer et al. 2018], but most approaches only reconstruct the geometry without the appearance of the objects in the scene. In contrast, our objective is to reconstruct a general dynamic scene, including its appearance, such that it can be rendered from novel viewpoints. This requires correctly reconstructing the

geometry, appearance and motion in the scene from the monocular RGB observations.

Recent Progress. Recent neural rendering approaches have shown impressive novel-view synthesis of general static scenes from multi-view input [Tewari et al. 2020]. These approaches represent scenes using trained neural networks and rely on less constraints about the type of scene, compared to traditional approaches. The closest prior work to our method is NeRF [Mildenhall et al. 2020a], which learns a continuous volume of the scene encoded in a neural network using multiple views from cameras with known extrinsic and intrinsic parameters. This volume stores the colors and opacities at every point, which enables rendering novel views using volumetric integration. This formulation has several advantages, as it does not assume any specific category of objects or any template mesh of the scene. However, NeRF assumes the scene to be static while most real world scenes are dynamic in nature. Neural Volumes [Lombardi et al. 2019] is another closely related approach that uses multiple views of a deforming scene to enable free viewpoint rendering. However, it uses a fixed-size voxel grid to represent the reconstruction of the scene, which restricts the resolution. In addition, it requires multi-view input for training, which limits the applicability to in-the-wild outdoor settings or existing monocular footage. Our new neural rendering approach instead targets the more challenging setting of using just a monocular video of a general dynamic scene. Because of the non-rigid deformations, each image of the video records a different, deformed state of the scene, which violates the constraints of standard neural rendering approaches. Our approach has to disentangle the observations in any image into a canonical scene and its deformations, without direct supervision on either component. This disentanglement allows for every image to (indirectly) supervise the canonical scene component, leading to correct reconstruction of scene representation and scene motion.

Our Approach. We tackle this problem using several innovations. As mentioned before, we represent the non-rigid scene as a combination of two components: (1) a canonical neural radiance field for capturing geometry and appearance and (2) the scene deformation field. The canonical volume is a static representation of the scene encoded as a Multi-Layered Perceptron (MLP), which is not directly supervised. This volume is deformed into each individual image using the estimated scene deformation. Due to the volumetric nature of the scene, we opt for space deformations, as opposed to surface deformation in mesh-based approaches. Specifically, the scene deformation is implemented as ray bending, where straight camera rays are allowed to deform non-rigidly. Ray bending is a complementary way of deforming the canonical volume: Instead of a point in the canonical volume deforming such that it lies on a straight camera ray, we deform the camera ray such that it hits the desired point in the canonical space. The ray bending is modeled using a MLP that takes point samples on the ray as well as a latent code for each image as input. Both the ray bending and the canonical scene MLPs are jointly trained using the monocular observations. Since the ray bending MLP deforms the entire space independent of any camera parameters, we can render the deforming volume from static or time-varying novel viewpoints after training. This

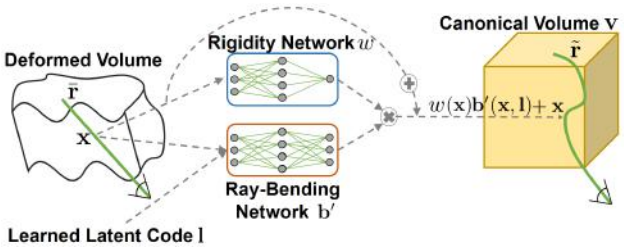


Fig. 2. Given only RGB input images $\{\hat{\mathbf{c}}_i\}_i$ of a dynamic scene, e.g., from a monocular video, and the associated camera parameters $\{\mathbf{R}_i, \mathbf{t}_i, \mathbf{K}_i\}_i$, NR-NeRF accumulates geometry and appearance information from all input images in a single static canonical neural radiance field volume $\mathbf{v}$ and models deformations as a per-time-step space warp of that volume. The space warping is realized as a ray-bending network $\mathbf{b}(\mathbf{x}, \mathbf{l}_i) = w(\mathbf{x})\mathbf{b}'(\mathbf{x}, \mathbf{l}_i)$ that regresses an offset for any point in 3D space, conditioned on an auto-decoded per-time-step deformation latent code $\mathbf{l}_i$. These offsets are applied to the straight input camera rays $\bar{\mathbf{r}}$ in order to bend them such that the input image is correctly reconstructed by integrating along the bent rays $\tilde{\mathbf{r}}$.

formulation also allows us to compute dense correspondences (via the canonical scene volume) between all time steps.

The ray bending MLP disentangles the geometry of the scene from the scene deformations. The disentanglement is an under-constrained problem, which we tackle with further innovations. Our method assigns a rigidity score to every point in the canonical volume, which allows for the deformations to not affect the static regions in the scene. This rigidity component is jointly learned without any direct supervision. We also introduce multiple regularizers as additional soft-constraints: A regularizer on the deformation magnitude of the *visible* deformations encourages only sparse deformations of the volume, and thus helps to constrain the canonical volume. An additional divergence regularizer preserves the local shape, thereby constraining the representation of *hidden* (partially occluded) regions that are not visible throughout the full video.

Our results show high-fidelity reconstruction and novel view synthesis for a wide range of non-rigid scenes. NR-NeRF also estimates, without direct supervision, rigidity scores and dense correspondences. Fig. 2 contains an overview of our method. To summarize, our main technical contributions are as follows:

- A free viewpoint rendering method, NR-NeRF, that only requires a monocular video of the dynamic scene, enabled by the disentanglement of the scene into a static canonical volume and per-frame scene deformation (Sec. 3). In NR-NeRF, camera poses at test time, i.e., for novel view synthesis, can differ significantly from the camera trajectory used for acquiring the input video. Moreover, as a side effect, we can extract dense correspondences relating arbitrary (input or novel) frames.
- A rigidity network which can segment the scene into non-rigid foreground and rigid background without being directly supervised (Sec. 3.2).
- Regularizers on the estimated deformations which constrain the problem by encouraging small volume preserving deformations (Sec. 3.3).

- Several extensions for handling of view dependence and multi-view data, and applications of our technique for scene editing such as motion exaggeration and dampening, foreground removal, and time interpolation (Secs. 3.4 and 4.6).

We compare NR-NeRF to several methods for neural novel view rendering and show results in Sec. 4. See our supplementary video for visualizations and Sec. 5 for a discussion.

2 RELATED WORK

Our approach closely relates to methods for the capture of static and dynamic scene geometry and appearance in 3D (Sec. 2.1) as well as image-based rendering and neural scene representations (Sec. 2.2).

2.1 4D Reconstruction and Novel Viewpoint Rendering

Early methods for image-based novel and free-viewpoint rendering combined traditional concepts of multi-view camera geometry, explicit vision-based 3D shape and appearance reconstruction, and classical computer graphics or image-based rendering. These methods are based on light fields [Buehler et al. 2001; Gortler et al. 1996; Levoy and Hanrahan 1996], multi-view stereo to capture dense depth maps [Zhang et al. 2003], layered depth images [Shade et al. 1998], or representations using 3D point clouds [Agarwal et al. 2011; Liu et al. 2010; Schonberger and Frahm 2016], meshes [Matsuyama et al. 2004; Tung et al. 2009] or surfels [Carceroni and Kutulakos 2002; Pfister et al. 2000; Waschbüsch et al. 2005] for dynamic scenes. Passive geometry capture often leads to artifacts in scenes with severe occlusions and view-dependent appearance. Also, capturing temporally coherent representations in this way is challenging.

More recently, the combination of multi-view stereo with fusion algorithms integrating implicit geometry over short time windows lead to improved results and short-term temporal coherence [Dou et al. 2016; Guo et al. 2017; Orts-Escolano et al. 2016]. By using active depth cameras and such fusion-type reconstruction, dynamic scene capture and novel viewpoint rendering from a low number of cameras or a single camera were shown [Huang et al. 2018; Tao et al. 2018; Yu et al. 2017; Yu et al. 2019]. Several algorithms use variants of shape-from-silhouette to approximate real scene geometry, such as visual hull reconstruction or visual hulls improved via multi-view photo-consistency in [Kutulakos and Seitz 2000; Starck et al. 2006]. While reconstruction is fast and feasible with fewer cameras, the coarse approximate geometry introduces rendering artifacts, and the reconstruction is usually limited to the separable foreground. Accurate and temporally coherent geometry is hard to capture in this way [Cagniart et al. 2010a,b]. Some approaches use 3D templates and combine vision-based reconstruction with appearance modelling to enable free-viewpoint video relighting, e.g., by estimating reflectance models under general lighting or under controlled light stage illumination [Guo et al. 2019; Li et al. 2013; Nagano et al. 2015; Theobalt et al. 2007].

The progress in RGB-D sensors has enabled depth map capture from a single camera. Such sensors can been used for 3D reconstruction and completion of rigid environments [Newcombe et al. 2011] and non-rigid objects [Innmann et al. 2016; Newcombe et al. 2015; Slavcheva et al. 2017a; Zollhöfer et al. 2014]. Other method classes allow capturing deformable geometry from sets of monocular views. Dense non-rigid structure from motion requires dense point tracks over input images, which are then factorized into camera poses and non-rigid 3D states for each view [Garg et al. 2013; Kumar et al. 2018; Sidhu et al. 2020]. The correspondences are usually obtained with dense optical flow techniques which makes them prone to occlusions and inaccuracies, and which can have a detrimental effect on the reconstructions. Monocular template-based methods do not assume dense correspondences and rely on a known 3D state of a deformable object (a 3D template), which is then tracked across time [Ngo et al. 2015; Perriollat et al. 2011; Xu et al. 2018; Yu et al. 2015], or a training dataset with multiple object states [Golyanik et al. 2018; Tretschk et al. 2020]. Templates ensure high temporal coherence, but limit the applications to specific scenes, e.g., videos of a specific surface or human, and they face difficulties in handling scenes with large deformations, fast motions, and complex interactions. Moreover, obtaining templates for complex objects and scenes is often non-trivial and requires specialized setups.

In contrast to the reviewed methods, our approach avoids explicit image-based 3D reconstruction. Moreover, we support arbitrary backgrounds whereas the discussed methods for monocular 3D reconstruction of deformable objects ignore it. Our approach enables free-viewpoint rendering of general deformable scenes with multiple objects and complex deformations with high visual fidelity and accuracy, and yet does not rely on templates, 2D correspondences and multi-view setups.

2.2 Neural Scene Representations and Neural Rendering

An emerging class of algorithms uses deep neural networks to augment or replace established graphics and vision concepts for reconstruction and novel-view rendering. The methods differ in the degree to which traditional reconstruction, scene representation and image formation are replaced with learned representations. Most recent work is designed for static scenes [Eslami et al. 2018; Flynn et al. 2019; Hedman et al. 2018; Meshry et al. 2019; Mildenhall et al. 2020b; Nguyen-Phuoc et al. 2018; Riegler and Koltun 2020; Sitzmann et al. 2019a,b]; methods handling dynamic scenes are in their infancy.

Several approaches address related problems to ours, such as generating images of humans in new poses [Balakrishnan et al. 2018; Ma et al. 2018; Neverova et al. 2018; Sarkar et al. 2020] or body reenactment from monocular videos [Chan et al. 2019]. Other methods combine explicit dynamic scene reconstruction and traditional graphics rendering with neural re-rendering [Kim et al. 2019, 2018; Martin Brualla et al. 2018]. Recently, Shysheya et al. [2019] proposed a neural rendering approach for human avatars with texture warping. The method is trained on annotated multi-view imagery and renders the target person at test time given the desired pose. Zhu et al. [2018] leverage geometric constraints and optical flow for synthesizing novel views of humans from a single image. Thies et al. [2019] combine neural textures with the classical graphics pipeline for novel view synthesis of static objects and monocular video re-rendering. Their technique requires a scene-specific geometric proxy which has to be reconstructed before the training. Neural Volumes (NVs) [Lombardi et al. 2019] learn object models which can be animated and rendered from novel views, given multi-view video data. NVs encode multi-view videos in compact latent

codes which are decoded into a semi-transparent volumetric grid with colors and transparencies. To generate the final image, the volume is then rendered by a differentiable ray marcher accumulating color and opacity for each pixel.

In contrast to all reviewed methods for neural representations of dynamic scenes, we require only a set of monocular views of a non-rigid scene as input and are able to render the scene from novel views or define an arbitrary trajectory of a virtual camera.

Non-Peer-Reviewed Reports. Since the intersection of neural scene representations and volumetric rendering has recently become a very active area of research with quickly evolving progress, several methods aimed at dynamic settings have been proposed concurrently to ours. We mention them only for completeness since they are not peer-reviewed and thus do not constitute prior work. Some methods extend neural radiance fields to deforming faces [Gafni et al. 2020; Gao et al. 2020; Wang et al. 2020]. Others focus on moving human bodies [Peng et al. 2020; Weng et al. 2020] or more general objects [Du et al. 2020; Li et al. 2020; Park et al. 2020; Pumarola et al. 2020; Xian et al. 2020]. Our method differs from these by tackling general, real-world dynamic scenes from monocular RGB observations and camera parameters only, without using any other supervisory scene information like optical flow or depth estimates.

3 METHOD

Our Non-Rigid Neural Radiance Field (NR-NeRF) approach takes as input a set of N RGB images $\{\hat{\mathbf{c}}_i\}_{i=0}^{N-1}$ of a non-rigid scene and their extrinsics $\{\mathbf{R}_i, \mathbf{t}_i\}_{i=0}^{N-1}$ and intrinsics $\{\mathbf{K}_i\}_{i=0}^{N-1}$. Non-rigid NeRF then finds a single canonical neural radiance volume that can be deformed via ray bending to correctly render each input view. Specifically, we collect appearance and geometry information in the static canonical volume $\mathbf{v}$ parametrized by weights θ. We model deformations by bending the straight rays sent out by a camera to obtain a deformed rendering of the canonical volume. This ray bending is implemented as a ray bending MLP $\mathbf{b}$ with weights ψ. It maps, conditioned on the current deformation, 3D points (for example sampled from the straight rays) to 3D positions in the canonical volume. The deformation conditioning takes the form of auto-decoded latent codes $\{\mathbf{l}_i\}_{i=0}^{N-1}$ for each image i.

3.1 Background: Neural Radiance Fields

We first recap NeRF [Mildenhall et al. 2020a] for rigid scenes. NeRF renders a 3D volume into an image by accumulating color (weighted by accumulated transmittance and density) along camera rays. The 3D volume is parametrized by an MLP $\mathbf{v}(\mathbf{x}, \mathbf{d}) = (\mathbf{c}, o)$ that regresses an RGB color $\mathbf{c} = \mathbf{c}(\mathbf{x}, \mathbf{d}) \in [0, 1]^3$ and an opacity $o = o(\mathbf{x}) \in [0, 1]$ for a point $\mathbf{x} \in \mathbb{R}^3$ on a ray with direction $\mathbf{d} \in \mathbb{R}^3$.

Let us consider a pixel (u, v) of an image $\hat{\mathbf{c}}_i$. In the case of a pinhole camera model, the associated ray $\mathbf{r}_{u,v}(j) = \mathbf{o} + j\mathbf{d}(u, v)$ can be calculated using $\mathbf{R}_i, \mathbf{t}_i$, and $\mathbf{K}_i$, which yield the ray origin $\mathbf{o} \in \mathbb{R}^3$ and ray direction $\mathbf{d}(u, v) \in \mathbb{R}^3$. We can then integrate along the ray from the near plane j_n to the far plane j_f of the camera frustum to obtain the final color $\mathbf{c}$ at (u, v):

$$\mathbf{c}(\mathbf{r}_{u,v}) = \int_{j_n}^{j_f} V(j) \cdot o(\mathbf{r}_{u,v}(j)) \cdot \mathbf{c}(\mathbf{r}_{u,v}(j), \mathbf{d}(u, v))\, dj \ , \quad (1)$$

with $V(j) = \exp(-\int_{j_n}^{j} o(\mathbf{r}_{u,v}(s))\, ds)$ being the accumulated transmittance along the ray from the near plane up to j.

In practice, the integrals are approximated by discrete samples $\mathbf{x}$ along the ray. Ideally, the samples would cluster around regions with large $V(j) \cdot o(\mathbf{r}_{u,v}(j))$ along the ray, since those regions contribute the most to the final color. However, since it is unknown a-priori where these regions are located, NeRF employs a coarse volume $\mathbf{v}_c$ with network weights θ_c and a fine volume $\mathbf{v}_f$ with network weights θ_f. Both volumes have the same architecture, but do not share weights: $\theta = \theta_c \dot{\cup} \theta_f$. When rendering a ray, the coarse volume is accessed first at uniformly distributed samples along the ray. These coarse samples are used to estimate the transmittance distribution V, from which fine samples are sampled. The fine volume is then evaluated at the combined set of coarse and fine sample points. We refer to the original paper for more details.

Adaptations for NR-NeRF. We assume Lambertian materials in the scene and thus remove the view-dependent layers of rigid NeRF, i.e., we set $\mathbf{c} = \mathbf{c}(\mathbf{x})$. Since each image corresponds to a different deformation of the volume in our non-rigid setting, we also learn a latent code for each time step, which is then used as input for the ray bending network which parameterizes scene deformations. The weights of this network as well as the latent codes are shared for the coarse and fine volumes.

3.2 Deformation Model

The original NeRF method [Mildenhall et al. 2020a] assumes static scenes and cannot handle non-rigid scenes. A naïve approach to modeling deformations in the NeRF framework would be to condition the volume on the deformation, e.g., by conditioning it on time or a deformation latent code. We explore the latter option in the experiments in Sec. 4.4, since, unlike a simple time conditioning, it theoretically enables the optimization to use the same conditioning for the same deformation even if it occurs at different time steps. As we will show, this naïve approach only leads to satisfying results when reconstructing the input camera path, but gives implausible results for novel view synthesis. Instead, we explicitly model the consistency of geometry and appearance across time by disentangling them from the deformation.

We accumulate geometry and appearance from all frames into a single, non-deforming canonical volume. Since we employ a volumetric representation, we cannot simply adopt standard surface deformation models like skeleton skinning to model the non-rigid deformations present in our scenes. Instead, we employ general space (or coordinate) warping on top of the static canonical volume. In our setting, we apply the space warping to the camera rays and we can hence interpret it as ray bending.

For an input image $\hat{\mathbf{c}}_i$ at training time, we want to render the canonical volume such that the image is reproduced. To that end, we need to un-do the deformation of the specific time step i by mapping the camera rays to the deformation-independent canonical volume. We first send out straight rays from the input camera, which is parametrized by $\mathbf{R}_i, \mathbf{t}_i$, and $\mathbf{K}_i$. In order to account for the deformation, we then bend the straight rays such that sampling and subsequently rendering the canonical volume along the bent

rays yields $\hat{\mathbf{c}}_i$. We choose a very unrestricted parametrization of the ray bending, namely an MLP.

Specifically, we implement ray bending as a ray bending network $\mathbf{b}(\mathbf{x}, \mathbf{l}_i) \in \mathbb{R}^3$. For a point $\mathbf{x}$, for example lying on a straight ray, the network regresses an offset under a deformation represented by $\mathbf{l}_i$. The offset is then added to $\mathbf{x}$, thus bending the ray. Finally, we pass the new, bent ray point to the canonical volume, that is: $(\mathbf{c}, o) = \mathbf{v}(\mathbf{x} + \mathbf{b}(\mathbf{x}, \mathbf{l}_i))$. Note that $\mathbf{v}$ is not conditioned on $\mathbf{l}_i$, which leads to the disentanglement of deformation ($\mathbf{b}$ and $\mathbf{l}_i$) from geometry and appearance ($\mathbf{v}$). We denote the bent version of the straight ray $\bar{\mathbf{r}}$ as $\tilde{\mathbf{r}}_{\mathbf{l}_i}(j) = \bar{\mathbf{r}}(j) + \mathbf{b}(\bar{\mathbf{r}}(j), \mathbf{l}_i)$.

Rigidity Network. However, we find that rigid parts of the scene are insufficiently constrained by this formulation. We reformulate $\mathbf{b}(\mathbf{x}, \mathbf{l}_i) \in \mathbb{R}^3$ as the product of a raw offset $\mathbf{b}'(\mathbf{x}, \mathbf{l}_i)$ and a rigidity mask $w(\mathbf{x}) \in [0, 1]$, i.e., $\mathbf{b}(\mathbf{x}, \mathbf{l}_i) = w(\mathbf{x})\mathbf{b}'(\mathbf{x}, \mathbf{l}_i)$. For rigid objects, we want to prevent deformations and hence desire $w(\mathbf{x}) = 0$, while for non-rigid objects, we want $w(\mathbf{x}) > 0$. This makes it easier for $\mathbf{b}'$ to focus on the non-rigid parts of the scene, which change over time, since rigid parts can get masked out by the rigidity network w, which is jointly trained. Because the rigidity network is not conditioned on the latent code $\mathbf{l}_i$, it is forced to share knowledge about the rigidity of regions in the scene across time steps, which also ensures that parts of the rigid background that can be unregularized at certain time steps are nonetheless reconstructed at all time steps without any deformation.

3.3 Losses

With the architecture fully specified, the next step is to optimize all parameters $((\theta, \psi, \{\mathbf{l}_i\}_i)$ jointly. While the network weights are optimized as usual, the latent codes $\mathbf{l}_i$ are auto-decoded, i.e., they are treated as free variables that are directly optimized for, similar to network weights, instead of being regressed. This is based on the auto-decoding framework used in DeepSDF and earlier works [Park et al. 2019; Tan and Mayrovouniotis 1995].

Notation. For ease of presentation, we consider a single time step i and a single straight ray $\bar{\mathbf{r}}$ with coarse ray points $\bar{C} = \{\bar{\mathbf{r}}(j)\}_{j \in C}$ for a set C of uniformly sampled $j \in [j_n, j_f]$ and fine ray points $\bar{F} = \{\bar{\mathbf{r}}(j)\}_{j \in F}$ for a set F of importance sampled j. For a latent code $\mathbf{l}$, the bent ray $\tilde{\mathbf{r}}_{\mathbf{l}}$ gives $\tilde{C} = \{\tilde{\mathbf{r}}_{\mathbf{l}}(j)\}_{j \in C}$ and $\tilde{F} = \{\tilde{\mathbf{r}}_{\mathbf{l}}(j)\}_{j \in F}$. The actual training uses a batch of randomly chosen rays from the training images.

Reconstruction Loss. We adapt the data term from NeRF to our non-rigid setting as follows:

$$L_{data} = \|\mathbf{c}_c(\tilde{C}) - \hat{\mathbf{c}}(\mathbf{r})\|_2^2 + \|\mathbf{c}_f(\tilde{C} \cup \tilde{F}) - \hat{\mathbf{c}}(\mathbf{r})\|_2^2 \ , \tag{2}$$

where $\hat{\mathbf{c}}(\mathbf{r})$ is the ground-truth color of the pixel and $\mathbf{c}(S)$ is the estimated ray color on the set S of discrete ray points.

While this reconstruction loss yields good results for input reconstruction, i.e., along the space-time camera trajectory of the input recording, we show later in Sec. 4.3 that it leads to undesirable renderings for novel views. Since our problem setting is more underconstrained than NeRF's, we find it necessary to regularize the bending of rays with further priors.

Offsets Loss. We regularize the offsets with a loss on their magnitude. Since we want air, i.e., visually unoccupied space, to be compressible and not hinder the optimization, we weigh the loss at each sample point by its opacity. However, this would still apply a high weight to completely occluded points along the ray, which leads to severe artifacts when rendering novel views. We thus additionally weigh by visibility:

$$L_{naïve\ offsets} = \frac{1}{|C|} \sum_{j \in C} \alpha_j \cdot (\|\mathbf{b}(\bar{\mathbf{r}}(j), \mathbf{l})\|_2^{2 - w(\tilde{\mathbf{r}}(j))}, \tag{3}$$

where we weigh each point by its visibility and occupancy $\alpha_j = V(j) \cdot o(\tilde{\mathbf{r}}(j))$. We do not back-propagate into α_j.

However, as we show in the experimental section, we find that applying the offsets loss to the masked offsets leads to an unstable background during novel view synthesis. We hypothesize that this is due to the multiplicative ambiguity between unmasked offsets and rigidity mask. We find that applying the loss to the regressed rigidity mask and raw offsets separately works better:

$$L_{offsets} = \frac{1}{|C|} \sum_{j \in C} \alpha_j \cdot \left(\|\mathbf{b}'(\bar{\mathbf{r}}(j), \mathbf{l})\|_2^{2 - w(\tilde{\mathbf{r}}(j))} + \omega_{\text{rigidity}} w(\bar{\mathbf{r}}(j))\right) \ , \tag{4}$$

where we penalize $\mathbf{b}'$ instead of $\mathbf{b}$. The exponent of the first term has the following effect: For non-rigid objects with w closer to 1, it becomes an ℓ_1 loss, which has two advantages: (1) the gradient is independent of the magnitude of the offset, so unlike with an ℓ_2 loss, small and large offsets/motions are treated equally, and (2) relative to an ℓ_2 loss, it encourages sparsity in the offsets field, which fits our scenes. On the other hand, for rigid objects with w closer to 0, it becomes an ℓ_2 loss, which tampers off in its gradient magnitude as the offset magnitude approaches 0, preventing noisy gradients that an ℓ_1 loss would have.

Divergence Loss. Since the offsets loss only constrains visible areas, we introduce additional regularization of hidden areas. Inspired by local, isometric shape preservation from computer graphics, like as-rigid-as-possible regularization for surfaces [Igarashi et al. 2005; Sorkine and Alexa 2007] or volume preservation for volumes [Slavcheva et al. 2017b], we seek to preserve the local shape after deformation. To that end, we propose to regularize the absolute value of the divergence of the offsets field. The Helmholtz decomposition [Bhatia et al. 2012] allows to split any twice-differentiable 3D vector field on a bounded domain into a sum of a rotation-free vector field and a divergence-free vector field. Thus, by penalizing the divergence, we encourage the vector field to be composed primarily of translations and rotations, effectively preserving volume. The divergence loss is:

$$L_{divergence} = \frac{1}{|C|} \sum_{j \in C} w'_j \cdot |\text{div}(\mathbf{b}(\bar{\mathbf{r}}(j), \mathbf{l}))|^2 \ , \tag{5}$$

where $w'_j = o(\tilde{\mathbf{r}}(j))$, which we do not back-propagate into, and the divergence div is the divergence of $\mathbf{b}$ with respect to the position $\bar{\mathbf{r}}(j)$. We choose an ℓ_2 loss because its gradient magnitude goes to 0 for a divergence close to 0.

We employ FFJORD's [Grathwohl et al. 2018] fast, approximate, unbiased divergence estimation, which reduces the computational

cost of the divergence estimation compared to an exact computation by a factor of three in our case. The divergence is defined as:

$$\operatorname{div}(\mathbf{b}(\mathbf{x})) = \operatorname{Tr}(\frac{\mathrm{d}\mathbf{b}(\mathbf{x})}{\mathrm{d}\mathbf{x}}) = \frac{\partial \mathbf{b}(\mathbf{x})_x}{\partial x} + \frac{\partial \mathbf{b}(\mathbf{x})_y}{\partial y} + \frac{\partial \mathbf{b}(\mathbf{x})_z}{\partial z} \quad , \tag{6}$$

where $\mathbf{b}(\mathbf{x})_k \in \mathbb{R}$ is the kth-component of $\mathbf{b}(\mathbf{x}) \in \mathbb{R}^3$, Tr is the trace operator, and $\frac{\mathrm{d}\mathbf{b}(\mathbf{x})}{\mathrm{d}\mathbf{x}}$ is the 3×3 Jacobian matrix. Naïvely computing the divergence with PyTorch's automatic differentiation requires three backward passes, one for each term of the sum. However, the authors of FFJORD [Grathwohl et al. 2018] suggest using Hutchinson's trace estimator [Hutchinson 1989] instead:

$$\operatorname{Tr}(\mathbf{A}) = \mathbb{E}_{\mathbf{e}}[\mathbf{e}^T \mathbf{A} \mathbf{e}] \quad . \tag{7}$$

Here, $\mathbf{e}$ is Gaussian-distributed. The single-sample Monte-Carlo estimator implied by this expectation can be computed with a single backward pass.

Full Loss. We combine all losses to obtain the full loss function:

$$L = L_{\text{data}} + \omega_{\text{offsets}} L_{\text{offsets}} + \omega_{\text{divergence}} L_{\text{divergence}} \quad , \tag{8}$$

where the weights ω_{rigidity}, ω_{offsets}, and ω_{offsets} are scene-specific.

Note that we do not exploit temporal information, e.g., in an explicit temporal term, in our formulation as we found that our approach already produces temporally-stable results without such a term. Further investigation thereof can be explored in future work.

3.4 Extensions

We can extend our approach easily to work with multi-view data and view-dependent effects and will show example results for each in Sec. 4.5.

Multi-View Data. Our approach naturally handles multi-view data. Instead of each image being associated with its own time step and hence latent code, images taken at the same time step share the same latent code. This ensures that the canonical volume deforms consistently within each time step.

View Dependence. We can optionally add view-dependent effects, like specularities, into our model. However, determining the view direction or ray direction is not as trivial as for the straight rays. Instead, we need to calculate the direction in which the bent ray passes through a point in the canonical volume. We consider two options of doing so: exact and slower, or approximate and faster.

Exact: We obtain the direction of the bent ray $\tilde{\mathbf{r}}$ at a point $\tilde{\mathbf{r}}(j)$ via the chain rule as $\nabla_j \tilde{\mathbf{r}}(j) = \frac{\partial \tilde{\mathbf{r}}(j)}{\partial \tilde{\mathbf{r}}(j)} \cdot \frac{\partial \tilde{\mathbf{r}}(j)}{\partial j} = J \cdot \mathbf{d}$, where J is the 3×3 Jacobian and $\mathbf{d}$ is the direction of the straight ray. We compute J via three backward passes, which is computationally expensive.

Approximate: To reduce computation, we can approximate the direction at the ray sample via finite differences as the normalized difference vector between the current point $\tilde{\mathbf{r}}(j)$ and the previous point $\tilde{\mathbf{r}}(j-1)$ along the bent ray (which is closer to the camera).

3.5 Training Details

We initialize $\{\mathbf{l}_i\}_i$ to zero vectors. For implementing the radiance field, we use the same architecture as in NeRF [Mildenhall et al. 2020a]. The ray bending network is a 5-layer MLP with 64 hidden dimensions and ReLU activations, the last layer of which is initialized with all weights set to zero. The rigidity network is a 3-layer MLP with 32 hidden dimensions and ReLU activations, with the last layer initialized to zeros. The output of the last layer of the rigidity network is passed though a tanh activation function and then shifted and rescaled to lie in $[0, 1]$. We train usually for 200k iterations with a batch of 1k randomly sampled rays. At training and at test time, we use 64 coarse and 64 fine samples per ray in most cases. We use ADAM [Kingma and Ba 2014] and exponentially decay the learning rate to 10% from the initial $5 \cdot 10^{-4}$ over 250k iterations. For dark scenes, we found it necessary to introduce a warm-up phase that linearly increases the learning rate starting from $\frac{1}{20}$th of its original value over 1000 iterations. The latent codes are of the dimension 32. We train between six and seven hours on a single Quadro RTX 8000.

Scene-Specific Weights. Since we consider a variety of types of scenes and deformations, we find it necessary to use scene-specific weights for each loss term. We have found the following ranges to be sufficient for a wide range of scenarios: ω_{rigidity} lies in $[0.01, 0.001]$ and typically is 0.003, ω_{offsets} lies in $[60, 600]$ and typically is 600, and $\omega_{\text{divergence}}$ lies in $[1, 30]$ and typically is 3 or 10. In our experience, NR-NeRF is fairly insensitive to ω_{offsets}. Rather rigid objects benefit from higher $\omega_{\text{divergence}}$, while fairly non-rigid objects need lower $\omega_{\text{divergence}}$. Finally, we increase ω_{rigidity} whenever we find the background to be unstable. We start the training with each weight set to $\frac{1}{100}$th of its value, and then exponentially increase it until it reaches its full value at the end of training.

3.6 Implementation Details

Our code is based on a faithful PyTorch [Paszke et al. 2019] port [Yen-Chen 2020] of the official Tensorflow NeRF code [Mildenhall et al. 2020a]. We use the official FFJORD implementation [Grathwohl et al. 2018] to estimate Eq. 7. If the camera extrinsics and intrinsics are not given, we estimate them using the Structure-from-Motion (SfM) implementation of COLMAP [Schönberger and Frahm 2016; Schönberger et al. 2016]. We find COLMAP to be quite robust to non-rigid 'outliers'. As we are interested in estimating smooth deformations, we only apply positional encoding to the input of the canonical NeRF volume, not to the input of the ray bending network. We will make our source code available.

4 RESULTS

We first provide information on data capture in Sec. 4.1. We then present qualitative results of our method by rendering into input and novel views in Sec. 4.2, and we visualize the estimated rigidity scores and correspondences. Turning from the outputs of our method to its inner workings, Sec. 4.3 investigates the crucial design choices we made to improve novel view quality. We conclude the evaluation of our approach in Sec. 4.4, where we compare our method to prior work and a baseline approach. Having established the working of our method, we explore natural extensions to multi-view data and view-dependent effects in Sec. 4.5. Finally, we show simple scene-editing results in Sec. 4.6.

4.1 Data

We show results on a variety of scenes recorded with three different cameras: the Kinect Azure, a Blackmagic, and a phone camera. Since the RGB camera of the Kinect Azure exhibits strong radial distortions along the image border, we use the manufacturer-provided intrinsics and distortion parameters to undistort the recorded RGB images beforehand. We extract frames at 5 fps from the recordings, such that scenes usually consist of 80 to 300 images, at resolutions of 480×270 (Blackmagic, and Sony XZ2) or 512×384 (Kinect Azure).

4.2 Qualitative Results

We present qualitative results of NR-NeRF by rendering the scene from both the input views and novel views. We also visualize the additional outputs of our method, namely the rigidity scores and correspondences. For many more qualitative results, we refer to the supplemental video.

Input Reconstruction and Novel View Synthesis. Fig. 3 gives an overview over the input reconstruction and novel view synthesis capabilities of our method. As the center column shows, the input is reconstructed faithfully. This enables high-quality novel view synthesis, example results can be found in the third column. We can freely move the camera in areas around the original camera paths and specify the time index.

Rigidity. NR-NeRF estimates rigidity scores without supervision in order to improve background stability in novel-view renderings. We show examples of these rigidity scores in Fig. 4 and find that the background is consistently scored as very rigid while foreground is correctly estimated to be rather non-rigid. In order to visualize the estimated rigidity, we need to determine the rigidity of the ray associated with a pixel. We choose to define the rigidity of such a ray as the rigidity of the point j closest to an accumulated weight $\sum_{k=0}^{j-1} \alpha_k$ of 0.5, *i.e.,* closest to the median. In practice, this usually gives us the rigidity at the first visible surface along the ray.

Correspondences. Another side effect of our proposed approach is the ability to estimate consistent dense 3D correspondences into the canonical model across different camera views and time steps. Fig. 4 shows examples. To visualize this, we treat the canonical volume as an RGB cube, i.e., we treat the xyz coordinate in canonical space as an RGB color. Since this would result in very smooth colors, we split the canonical volume into a voxel grid of 100^3 RGB cubes beforehand. We pick the ray point that determines the pixel color similar to the rigidity visualization.

Canonical Volume. Since the canonical volume is not supervised directly, it is conceivable that it could have baked-in deformations. The last row of Fig. 4 contains renderings of the canonical volume without any ray bending applied. The canonical volume is a plausible state of the scene and does not show baked-in deformations. We thus find it to be sufficient to bias the optimization towards a desirable canonical volume by initializing the ray bending network to an identity map and by our regularization losses.

4.3 Ablation Study

After a look at the outputs of NR-NeRF in the previous section, we now turn to the internal workings of our approach. Specifically, since we aim for convincing novel-view renderings, we take a closer look at the qualitative impact of some of our design choices on novel view results.

Setup. We investigate the necessity of all regularization losses by removing each loss individually and all of them at once. Next, we remove the rigidity network to see the impact on background stability. Finally, we determine whether applying the offsets loss separately on both the regressed rigidity and the unmasked offsets, *i.e.* $L_{offsets}$ in our method, or directly on the masked offsets, $L_{naïve\ offsets}$, works better.

Results. We find that the divergence loss is crucial for stable deformations of the non-rigid objects in the scene, see Fig. 5. The remaining design choices mainly improve the stability of the background as Fig. 6 shows. The supplemental video contains video examples that highlight the instability in these cases.

4.4 Comparisons

Having only considered our method in isolation so far, we compare NR-NeRF to prior work and a baseline in this subsection. We first introduce the prior work against which we compare and a baseline, then explain our training/test split, and finally show qualitative and quantitative comparison results.

Prior Work and Baseline. We start with the trivial baseline of rigid NeRF [Mildenhall et al. 2020a], which cannot handle dynamic scenes. We consider two variants: view-dependent rigid NeRF, as in the original method [Mildenhall et al. 2020a], and view-independent rigid NeRF, where we remove the view-direction conditioning.

We next introduce *naïve NR-NeRF*, which adds naïve support for dynamic scenes to rigid NeRF: We condition the neural radiance fields volume on the latent code. Thus, for latent code $\mathbf{l}_i$, we have $(\mathbf{c}, o) = \mathbf{v}(\mathbf{x}, \mathbf{l}_i)$. This allows the neural radiance fields volume to output time-varying color and occupancy. Unlike NR-NeRF's ray bending, naïve NR-NeRF does not have an explicit, separate deformation model. Instead, the volume needs to account for appearance, geometry and deformation at once. Note that for test images i, we do backpropagate gradients into the corresponding latent code $\mathbf{l}_i$.

Finally, we compare to Neural Volumes [Lombardi et al. 2019], for which we use the official code release. We use the standard settings as a starting point, but set the number of training iterations to $100k$, which leads to a training time of about two days on an RTX8000 GPU. Neural Volumes uses an image encoder to regress a latent code that conditions the geometry, appearance and deformation regression on the current time step. Since this design assumes a multi-view setup, we need to adapt it to our monocular setting. Instead of picking three fixed camera views that are always input into the encoder, we input the single image of the current time step. Since we do not have access to a background image, we set the estimated background image to an all-black image. We furthermore consider two variations: (1) following the original Neural Volumes method, the geometry and appearance template is conditioned on the latent code (*NV*), and (2) the geometry and appearance template

Fig. 3. The input (left) is reconstructed by NR-NeRF (middle) and rendered into a novel view (right).

is independent of the latent code (*modified NV*). In the latter case, the latent code only conditions the warp field, which is similar to our method.

Training/Test Split. For quantitative evaluation, we require a test set. In this subsection, we split the images into training and test images by partitioning the temporally-ordered images into consecutive blocks, each of length 16. The first twelve images of each block are used as training data, while the remaining four are used for testing. In our setting, test images still require corresponding latent codes to represent the deformations. Therefore, we treat test images like training images except that we do not backpropagate into the canonical volume or the ray bending network. However, we do use gradients from test images to optimize the corresponding latent codes. Note that all other results shown in this paper, outside of this subsection, treat all images as training images since we train scene-specific networks. Furthermore, qualitative baseline results in the supplemental video show both training and test time steps.

Input Reconstruction. We first consider input reconstruction quality on the training set to verify the plausibility of the learned representations. See Fig. 7 and the supplemental video for results. We find that naïve NR-NeRF and both variants of Neural Volumes perform very well on this task, similar to our method. However, rigid NeRF's not accounting for deformations leads to blur. Interestingly, rigid NeRF can exploit view dependence to simulate deformations as view-dependent effects in sequences where camera view and deformation are correlated (*e.g.,* when different camera views are not revisited). We refer to the supplemental video for examples.

We next evaluate novel-view performance qualitatively and quantitatively on the test set, and we look at background stability as one of the major difficulties that methods face when rendering novel views.

Novel View Synthesis: Qualitative. Fig. 7 and the supplemental video contain novel-view results of all methods. Both versions of Neural Volumes give implausible results that are in some cases only barely recognizable. The two rigid NeRF variants show blurry, static results similar to the training reconstruction results earlier. Naïve NR-NeRF comes closest to the quality of our novel-view renderings. While the still images in Fig. 7 show some undesirable artifacts like blurrier (sixth row) or less stable (fourth row) results compared to ours, we refer to the supplemental video to see the temporal inconsistencies that come up in the naïve formulation.

Novel View Synthesis: Quantitative. After the qualitative overview, we now evaluate the novel-view results of the methods considered quantitatively. We use the same three metrics as NeRF [Mildenhall et al. 2020a]. We use PSNR and SSIM [Zhou Wang et al. 2004] as conventional metrics for image similarity, where higher is better. In addition, we use a learned perceptual metric, LPIPS [Zhang et al. 2018], where lower is better. Tab. 4.4 contains quantitative results of our method, the two rigid NeRF versions, naïve NR-NeRF, and two variants of Neural Volumes. We report averages across all our scenes. Our method (and view-dependent variants thereof) obtain the best and second-best scores on SSIM and LPIPS, and the second-best on PSNR. Naïve NR-NeRF gets the best PSNR value of all methods. As we saw in the input reconstruction results, naïve NR-NeRF is competitive for settings that are close to input views, as is the case for our test sets. We thus next evaluate background stability for a more challenging novel-view scenario where the camera stays fixed throughout the rendering.

Novel View Synthesis: Background Stability. While a moving camera during rendering can obfuscate background instability, we found that stabilizing the background for fixed novel-view renderings is difficult. We thus evaluate this challenging task here. In Fig. 8, we compare the background stability of our method, naïve NR-NeRF, and the Neural Volumes variants. We exclude rigid NeRF since it is static by design. We find that our method leads to significantly more stable background synthesis than the other methods, and we refer to the supplemental video for more qualitative results of this.

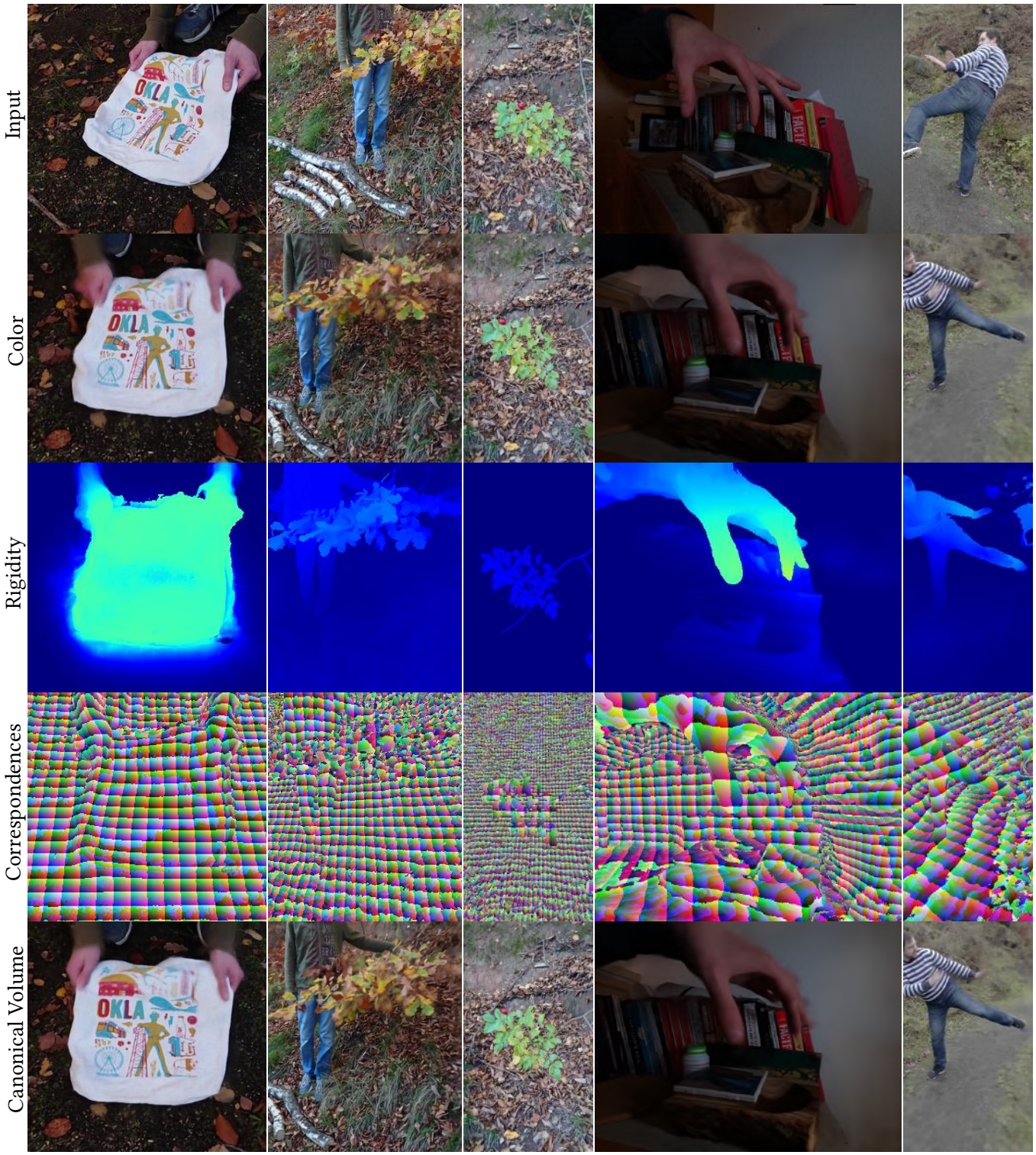


Fig. 4. NR-NeRF can render a deformed state captured at a certain time into a novel view. We visualize here this novel-view rendering and additional modalities as seen from the novel view, namely rigidity scores, correspondences, and the canonical volume.

	Ours	Ours (approx. view)	Ours (exact view)	Naïve NR-NeRF	Rigid (view-dependent)	Rigid (not view-dependent)	NV	NV (modified)
PSNR	24.70	*25.15*	25.07	**25.83**	22.24	21.88	14.13	14.10
SSIM [Zhou Wang et al. 2004]	0.758	**0.766**	*0.765*	0.738	0.662	0.659	0.259	0.263
LPIPS [Zhang et al. 2018]	0.197	*0.191*	**0.190**	0.226	0.309	0.313	0.580	0.583

Table 1. Quantitative Results Averaged Across Scenes. We evaluate our method (1) without view conditioning, (2) with approximate view conditioning, and (3) with exact view conditioning, naïve NR-NeRF, rigid NeRF [Mildenhall et al. 2020a] (1) with view conditioning and (2) without view conditioning, and Neural Volumes [Lombardi et al. 2019] (1) without and (2) with modifications. For PSNR and SSIM, higher is better. For LPIPS, lower is better.

Without $L_{divergence}$ No regularization Ours

Fig. 5. Ablation Study. We render the input scenes into novel views to determine the stability of the non-rigid objects. We show the results of removing the divergence loss,all three regularization losses, and none of the losses.

4.5 Extensions

We now show results of the extensions described in Sec. 3.4.

Multi-View Data. Although we mainly work with monocular data, we can use multi-view data to investigate the upper quality bound of our approach under ideal real-world conditions. We use a multi-view dataset that has 16 camera pairs evenly distributed around the scene, which sufficiently constrains the optimization such that we find the training to not need any regularization losses. We train at the original resolution of 5120×3840 for 2 million training iterations with 4096 rays per batch and 256 coarse and 128 fine samples. These highest-quality settings lead to a training time of 11 days on 4 RTX8000 GPUs, and a rendering time of about 10 minutes per frame on the same hardware. See Fig. 12 and the supplemental video for results on five consecutive time steps.

View Dependence. We can extend NR-NeRF to optionally model view-dependent effects. On multi-view data, conditioning on the viewing direction reduces the presence of subtle, smoke-like artifacts, which the canonical volume typically employs to model view-dependent effects without view conditioning. See Fig. 12 for results. For quantitative results on monocular sequences, see Tab. 4.4. However, as Fig. 9 shows, we find our formulation to lead to artifacts in some cases. We hypothesize that the combination of both significant motion and novel views significantly different from input views is too underconstrained for view-dependent effects. For example, similar to rigid NeRF results in the supplemental video, non-rigid NeRF might overfit to subtle correlations between deformation and camera position at training time. However, we want to emphasize that better formulations and regularization in future work may make view-dependent effects work in these challenging scenarios.

4.6 Simple Scene Editing

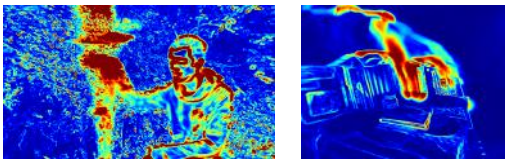
We can manipulate the learned model in several simple ways: temporal super-sampling, deformation exaggeration or dampening, foreground removal, and forced background stabilization.

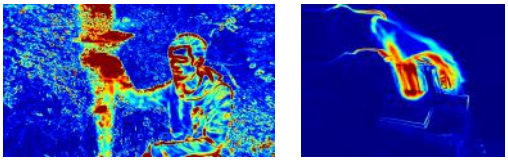
Time Interpolation. We can linearly interpolate between consecutive time steps to enable temporal super-resolution since NR-NeRF optimizes a latent code $\mathbf{l}_i$ for every time step i. We refer to the supplemental video for an example on multi-view data. While the person moves very smoothly in general, the interpolation leads to small artifacts on the belly.

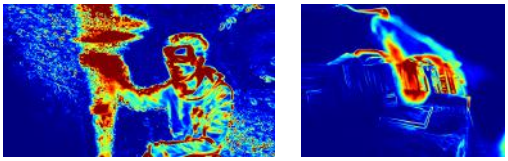
Deformation Exaggeration/Dampening. We can manipulate the deformation even further. Specifically, we can exaggerate or dampen deformations relative to the canonical model by scaling all offsets with a constant $m \in \mathbb{R}$: $(\mathbf{c}, o) = \mathbf{v}(\mathbf{x} + m\mathbf{b}(\mathbf{x}, \mathbf{l}_i))$. Fig. 10 and the supplemental video contain examples.

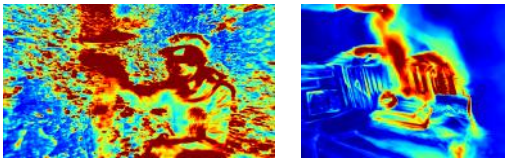
Foreground Removal. Our learned representation enables us to remove a potentially occluding non-rigid object from the foreground, leaving only the unoccluded background. Assuming the rigidity network assigns high scores to non-rigid objects and low scores to rigid (background) objects, we can threshold them at test time to segment the canonical volume into rigid and non-rigid parts. We can then isolate the rigid background, effectively removing the occluding non-rigid object. See Fig. 11 and the supplemental video for examples.

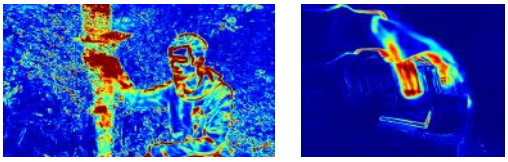
Forced Background Stabilization. Since we do not require any pre-computed foreground-background segmentation, NR-NeRF has to assign rigidity scores without supervision. Occasionally, this insufficiently constrains the background and leads to small motion. We can fix this in some cases by enforcing a stable background at test time Specifically, we threshold the regressed score at some value r_{min} and replace it with 0 if it is below r_{min}. If the rigid background has sufficiently small scores assigned to it relative to the non-rigid part of the scene, this forces the background to remain static for all time steps and views. For results, we refer to the supplemental video.

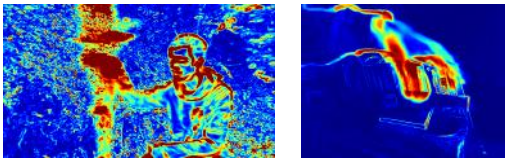

Without first term of offsets loss $L_{offsets}$


Without second term of offsets loss $L_{offsets}$


Without divergence loss $L_{divergence}$


Without any regularization loss


Without rigidity network w


With $L_{naïve\ offsets}$

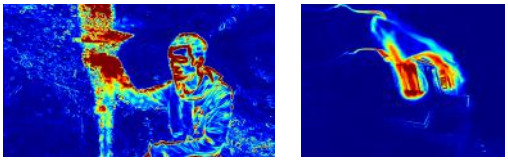

Ours

Fig. 6. Ablation Study. We determine the impact of our main design choices on background stability. To that end, we render each input scene into a fixed novel view and compute the standard deviation of each pixel's color across all time steps. Higher standard deviation indicates color changes and hence instability of the rigid background. The results of NR-NeRF show the least instability. We multiply the resulting standard deviation by 10 for better visualization and average across color channels.

5 LIMITATIONS AND FUTURE WORK

For simplicity, the discrete integration along the bent ray uses the interval lengths given by the straight ray. We do not expect this to cause noticeable issues since our scenes do not appear to lead to strong ray bending.

As we build on NeRF, our method is similarly slow. However, ray bending only leads to an increase of runtime by about 20%. Neural Sparse Voxel Fields [Liu et al. 2020] are a promising direction of speeding up NeRF-like approaches.

The background needs to be fairly close to the foreground, an issue we "inherit" from NeRF. Extending our method similar to NeRF++ [Zhang et al. 2020] to handle parts of the scene outside a certain foreground sphere separately would be an interesting direction for addressing this problem.

Since we use a deformation model that does not go from the canonical space to the deformed space, we cannot obtain exact correspondences between images captured at different time steps, but instead need to use a nearest neighbor/sample approximation. Future work could attempt to solve this issue by using an invertible one-to-one mapping for the space warping. Specifically, neural ODEs [Chen et al. 2018] would provide an interesting albeit slow alternative to our MLPs that would provide the opportunity for novel regularization terms.

We do not account for appearance changes that are due to deformation or lighting changes. For example, temporally changing shadowing in the input images is an issue, as Fig. 13 demonstrates.

Foreground removal can fail if a part of the foreground is entirely static, e.g., the food in Fig. 11, since the rigidity scores will mark it as rigid and hence background.

Rendering parts of the scene barely or not at all observed in the training data would not lead to realistic results.

The background needs to be static and dominant enough for SfM to estimate correct extrinsics.

Since our problem is severely under-constrained, we employ strong regularization, which leads to a trade-off between sharpness and stability on some scenes, as Fig. 14 shows.

6 CONCLUSION

We presented a method for free viewpoint rendering of a dynamic scene using just a monocular video as input. Several high-quality reconstruction and novel view synthesis results of general dynamic scenes, as well as unsupervised, yet plausible rigidity scores and dense 3D correspondences demonstrate the capabilities of the proposed method. Our results suggest that space warping in the form of ray bending is a promising deformation model for volumetric representations like NeRF. Furthermore, we have demonstrated that background instability, a problem also noted by concurrent work [Park et al. 2020], can be significantly mitigated in an unsupervised fashion by learning a rigidity mask for the offsets field. The sparsity and volume preserving regularizers we introduced help to better constrain the problem. The extensions to multi-view data and view dependence invite future work on more constraint settings to achieve higher quality. Although rather rudimentary, we have shown that NR-NeRF already enables several scene-editing tasks, and we look forward to further explorations in the direction

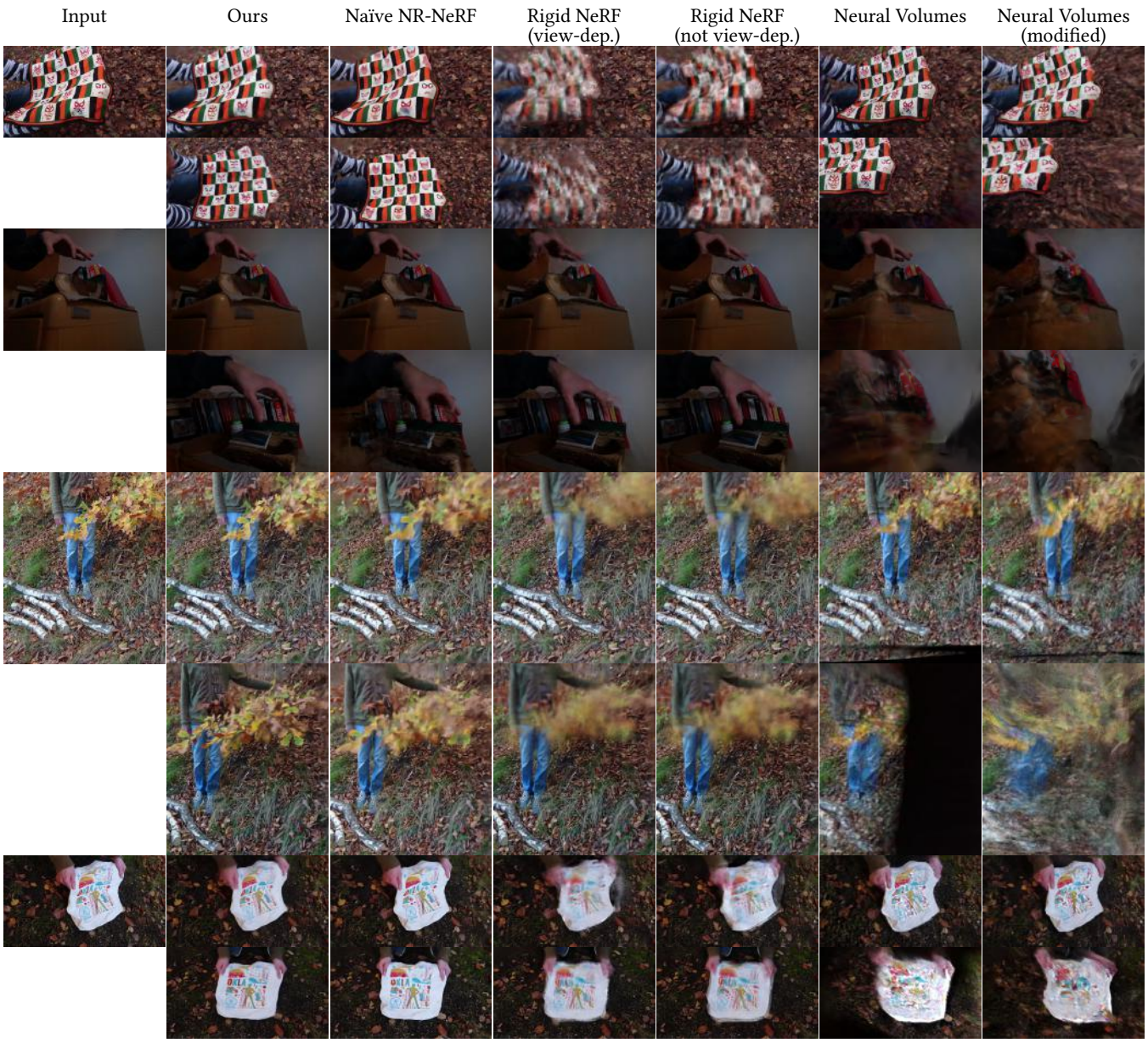


Fig. 7. We show one time step each from four sequences and compare input reconstruction quality (first row) and novel view synthesis quality (second row).

of editable neural representations. While several limitations exist as explained in Sec. 5, we hope that the technical contributions introduced will inspire future work.

Acknowledgements. All data capture and evaluation was done at MPII and VoluCap. Research conducted by Ayush Tewari, Vladislav Golyanik and Christian Theobalt at MPII was supported in part by the ERC Consolidator Grant 4DReply (770784). This work was also supported by a Facebook Reality Labs research grant. We thank VoluCap for providing the multi-view data.

REFERENCES

Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M. Seitz, and Richard Szeliski. 2011. Building Rome in a Day. *Commun. ACM* 54, 10 (2011), 105–112.

Guha Balakrishnan, Amy Zhao, Adrian V. Dalca, Frédo Durand, and John V. Guttag. 2018. Synthesizing Images of Humans in Unseen Poses. *Computer Vision and Pattern Recognition (CVPR)* (2018).

Harsh Bhatia, Gregory Norgard, Valerio Pascucci, and Peer-Timo Bremer. 2012. The Helmholtz-Hodge decomposition—a survey. *IEEE Transactions on Visualization and Computer Graphics (TVCG)* 19, 8 (2012), 1386–1404.

Chris Buehler, Michael Bosse, Leonard McMillan, Steven J. Gortler, and Michael F. Cohen. 2001. Unstructured lumigraph rendering. In *SIGGRAPH*.

Cedric Cagniart, Edmond Boyer, and Slobodan Ilic. 2010a. Free-form mesh tracking: A patch-based approach. *Computer Vision and Pattern Recognition (CVPR)* (2010), 1339–1346.

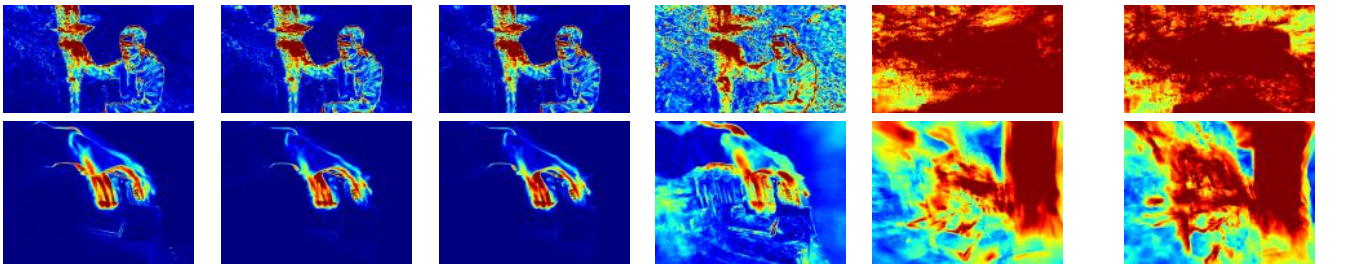


Fig. 8. We compare background stability. To that end, we render each input scene into a fixed novel view and compute the standard deviation of each pixel's color across test time steps. Higher standard deviation indicates color changes and hence instability of the rigid background. The results of NR-NeRF show the least instability. We multiply the resulting standard deviation by 10 for better visualization and average across color channels. We do not show rigid NeRF because it is perfectly static by construction.

Fig. 9. While NR-NeRF extended with view-dependent effects (approximate or exact) gives similar results to the default NR-NeRF for many monocular scenes, we sometimes observe artifacts for difficult novel views.

Cedric Cagniart, Edmond Boyer, and Slobodan Ilic. 2010b. Probabilistic Deformable Surface Tracking from Multiple Videos. In *European Conference on Computer Vision (ECCV)*.

Rodrigo L. Carceroni and Kiriakos N. Kutulakos. 2002. Multi-View Scene Capture by Surfel Sampling: From Video Streams to Non-Rigid 3D Motion, Shape and Reflectance. *International Journal of Computer Vision* 49, 2 (2002), 175–214.

Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A Efros. 2019. Everybody Dance Now. In *International Conference on Computer Vision (ICCV)*.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. 2018. Neural Ordinary Differential Equations. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Alvaro Collet, Ming Chuang, Pat Sweeney, Don Gillett, Dennis Evseev, David Calabrese, Hugues Hoppe, Adam Kirk, and Steve Sullivan. 2015. High-quality streamable free-viewpoint video. *ACM Transactions on Graphics (ToG)* 34, 4 (2015), 69.

Mingsong Dou, Sameh Khamis, Yury Degtyarev, Philip Davidson, Sean Ryan Fanello, Adarsh Kowdle, Sergio Orts Escolano, Christoph Rhemann, David Kim, Jonathan Taylor, and et al. 2016. Fusion4D: Real-Time Performance Capture of Challenging Scenes. *ACM Trans. Graph.* 35, 4 (2016).

Yilun Du, Yinan Zhang, Hong-Xing Yu, Joshua B. Tenenbaum, and Jiajun Wu. 2020. Neural Radiance Flow for 4D View Synthesis and Video Processing. *arXiv preprint arXiv:2012.09790* (2020).

Bernhard Egger, William A. P. Smith, Ayush Tewari, Stefanie Wuhrer, Michael Zollhoefer, Thabo Beeler, Florian Bernard, Timo Bolkart, Adam Kortylewski, Sami Romdhani, Christian Theobalt, Volker Blanz, and Thomas Vetter. 2020. 3D Morphable Face Models - Past, Present and Future. *ACM Transactions on Graphics* 39, 5 (Aug. 2020).

SM Ali Eslami, Danilo Jimenez Rezende, Frederic Besse, Fabio Viola, Ari S Morcos, Marta Garnelo, Avraham Ruderman, Andrei A Rusu, Ivo Danihelka, Karol Gregor, et al. 2018. Neural scene representation and rendering. *Science* 360, 6394 (2018), 1204–1210.

John Flynn, Michael Broxton, Paul Debevec, Matthew DuVall, Graham Fyffe, Ryan Overbeck, Noah Snavely, and Richard Tucker. 2019. DeepView: View Synthesis with Learned Gradient Descent. *International Conference on Computer Vision and Pattern Recognition (CVPR)* (2019).

Guy Gafni, Justus Thies, Michael Zollhöfer, and Matthias Nießner. 2020. Dynamic Neural Radiance Fields for Monocular 4D Facial Avatar Reconstruction. *arXiv e-prints* (2020).

Chen Gao, Yichang Shih, Wei-Sheng Lai, Chia-Kai Liang, and Jia-Bin Huang. 2020. Portrait Neural Radiance Fields from a Single Image. *arXiv e-prints* (2020). arXiv:2012.05903

Ravi Garg, Anastasios Roussos, and Lourdes Agapito. 2013. Dense Variational Reconstruction of Non-rigid Surfaces from Monocular Video. *Computer Vision and Pattern Recognition (CVPR)* (2013), 1272–1279.

Vladislav Golyanik, Soshi Shimada, Kiran Varanasi, and Didier Stricker. 2018. HDM-Net: Monocular Non-Rigid 3D Reconstruction with Learned Deformation Model. In *EuroVR*.

Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. 1996. The Lumigraph. In *SIGGRAPH*. 43–54.

Will Grathwohl, Ricky TQ Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud. 2018. Ffjord: Free-form continuous dynamics for scalable reversible generative models. *arXiv preprint arXiv:1810.01367* (2018).

Kaiwen Guo, Peter Lincoln, Philip Davidson, Jay Busch, Xueming Yu, Matt Whalen, Geoff Harvey, Sergio Orts-Escolano, Rohit Pandey, Jason Dourgarian, and et al. 2019. The Relightables: Volumetric Performance Capture of Humans with Realistic Relighting. *ACM Trans. Graph.* 38, 6 (2019).

Kaiwen Guo, Feng Xu, Tao Yu, Xiaoyang Liu, Qionghai Dai, and Yebin Liu. 2017. Real-Time Geometry, Albedo, and Motion Reconstruction Using a Single RGB-D Camera. *ACM Trans. Graph.* 36, 4 (2017).

Marc Habermann, Weipeng Xu, Michael Zollhöfer, Gerard Pons-Moll, and Christian Theobalt. 2019. LiveCap: Real-Time Human Performance Capture From Monocular Video. *ACM Trans. Graph.* 38, 2, Article 14 (March 2019), 17 pages. https://doi.org/10.1145/3311970

Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. 2018. Deep Blending for Free-viewpoint Image-based Rendering. *ACM Trans. Graph.* 37, 6, Article 257 (Dec. 2018), 15 pages. https://doi.org/10.1145/3272127.3275084

Zeng Huang, Tianye Li, Weikai Chen, Yajie Zhao, Jun Xing, Chloe Legendre, Linjie Luo, Chongyang Ma, and Hao Li. 2018. Deep Volumetric Video From Very Sparse Multi-view Performance Capture. In *European Conference on Computer Vision (ECCV)*. 351–369.

Michael F Hutchinson. 1989. A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines. *Communications in Statistics-Simulation and Computation* 18, 3 (1989), 1059–1076.

Takeo Igarashi, Tomer Moscovich, and John F Hughes. 2005. As-rigid-as-possible shape manipulation. *ACM transactions on Graphics (TOG)* 24, 3 (2005), 1134–1141.

Matthias Innmann, Michael Zollhöfer, Matthias Nießner, Christian Theobald, and Marc Stamminger. 2016. VolumeDeform: Real-Time Volumetric Non-rigid Reconstruction. In *European Conference on Computer Vision (ECCV)*.

Hyeongwoo Kim, Mohamed Elgharib, Hans-Peter Zollöfer, Michael Seidel, Thabo Beeler, Christian Richardt, and Christian Theobalt. 2019. Neural Style-Preserving Visual Dubbing. *ACM Transactions on Graphics (TOG)* 38, 6 (2019), 178:1–13.

Hyeongwoo Kim, Pablo Garrido, Ayush Tewari, Weipeng Xu, Justus Thies, Matthias Nießner, Patrick Pérez, Christian Richardt, Michael Zollöfer, and Christian Theobalt. 2018. Deep Video Portraits. *ACM Transactions on Graphics (TOG)* 37 (2018).

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

Muhammed Kocabas, Nikos Athanasiou, and Michael J. Black. 2020. VIBE: Video Inference for Human Body Pose and Shape Estimation. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 5252–5262.

Suryansh Kumar, Anoop Cherian, Yuchao Dai, and Hongdong Li. 2018. Scalable Dense Non-Rigid Structure-From-Motion: A Grassmannian Perspective. In *Computer Vision and Pattern Recognition (CVPR)*.

Kiriakos N. Kutulakos and Steven M. Seitz. 2000. A Theory of Shape by Space Carving. *International Journal of Computer Vision (IJCV)* 38 (2000), 199–218.

Marc Levoy and Pat Hanrahan. 1996. Light Field Rendering. In *SIGGRAPH*. 31–42.

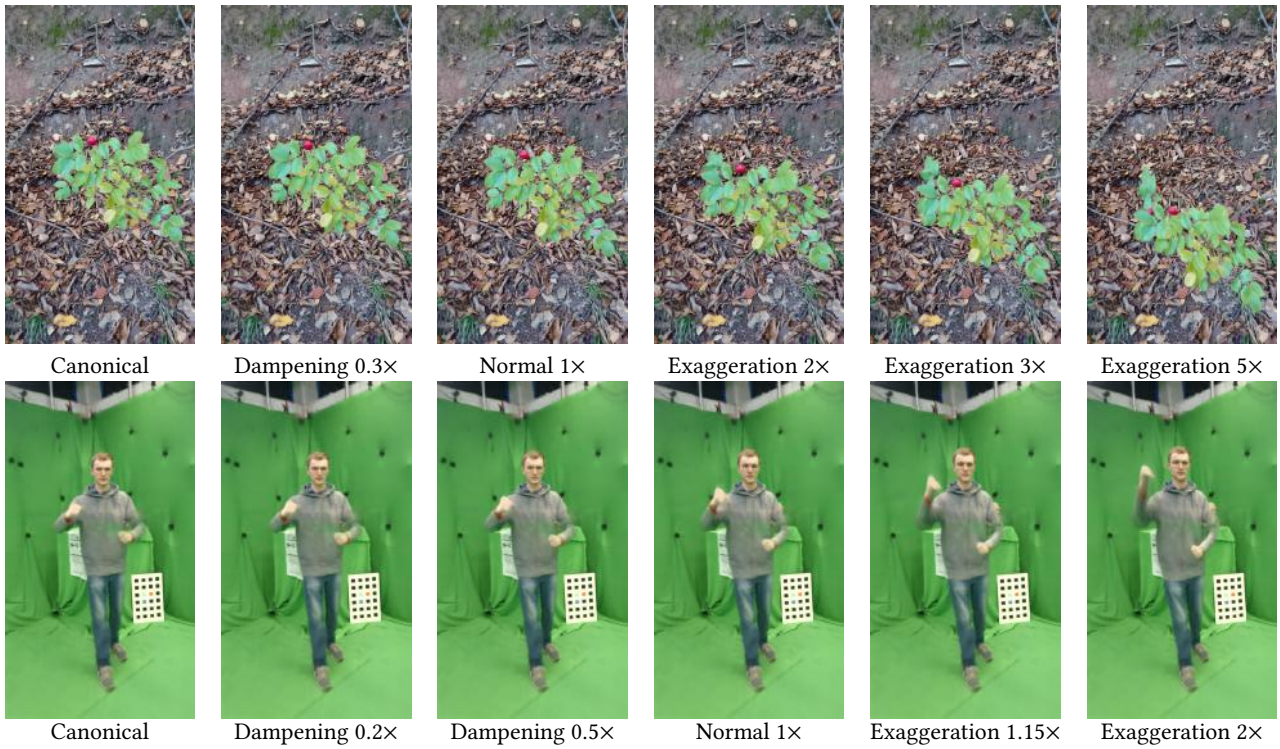


Fig. 10. We exaggerate or dampen the motion relative to the canonical model, and render the result into a novel view.

Fig. 11. Each image pair shows (left) the groundtruth input image and (right) a rendering without non-rigid foreground. NR-NeRF assigns rigidity scores to every point in space in an unsupervised manner. If all points of the rigid background are scored as more rigid than all points of the non-rigid foreground, we can remove the non-rigid foreground.

Guannan Li, Chenglei Wu, Carsten Stoll, Yebin Liu, Kiran Varanasi, Qionghai Dai, and Christian Theobalt. 2013. Capturing relightable human performances under general uncontrolled illumination. *Comput. Graph. Forum* 32, 2 (2013), 275–284.

Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. 2020. Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes. *arxiv* (2020).

Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. 2020. Neural sparse voxel fields. *Advances in Neural Information Processing Systems* 33 (2020).

Y. Liu, Q. Dai, and W. Xu. 2010. A Point-Cloud-Based Multiview Stereo Algorithm for Free-Viewpoint Video. *IEEE Transactions on Visualization and Computer Graphics (TVCG)* 16, 3 (2010), 407–418.

Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. 2019. Neural Volumes: Learning Dynamic Renderable Volumes from Images. *ACM Trans. Graph. (SIGGRAPH)* 38, 4 (2019).

Liqian Ma, Qianru Sun, Stamatios Georgoulis, Luc Van Gool, Bernt Schiele, and Mario Fritz. 2018. Disentangled Person Image Generation. *Computer Vision and Pattern Recognition (CVPR)* (2018).

Ricardo Martin Brualla, Peter Lincoln, Adarsh Kowdle, Christoph Rhemann, Dan Goldman, Cem Keskin, Steve Seitz, Shahram Izadi, Sean Fanello, Rohit Pandey, Shuoran Yang, Pavel Pidlypenskyi, Jonathan Taylor, Julien Valentin, Sameh Khamis, Philip Davidson, and Anastasia Tkach. 2018. LookinGood: Enhancing performance capture with real-time neural re-rendering. *ACM Transactions on Graphics* 37.

T. Matsuyama, Xiaojun Wu, T. Takai, and T. Wada. 2004. Real-time dynamic 3-D object shape reconstruction and high-fidelity texture mapping for 3-D video. *IEEE Transactions on Circuits and Systems for Video Technology* 14, 3 (2004), 357–369.

Moustafa Meshry, Dan B. Goldman, Sameh Khamis, Hugues Hoppe, Rohit Pandey, Noah Snavely, and Ricardo Martin-Brualla. 2019. Neural Rerendering in the Wild. In *Computer Vision and Pattern Recognition (CVPR)*.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020a. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *ECCV*.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020b. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *European Conference on Computer Vision (ECCV)*.

Graham Miller, Adrian Hilton, and Jonathan Starck. 2005. Interactive free-viewpoint video. In *IEEE European Conf. on Visual Media Production*. 50–59.

Koki Nagano, Graham Fyffe, Oleg Alexander, Jernej Barbiç, Hao Li, Abhijeet Ghosh, and Paul Debevec. 2015. Skin Microstructure Deformation with Displacement Map Convolution. *ACM Trans. Graph.* 34, 4 (2015).

Natalia Neverova, Riza Alp Güler, and Iasonas Kokkinos. 2018. Dense Pose Transfer. *ECCV* (2018).

Richard A. Newcombe, Dieter Fox, and Steven M. Seitz. 2015. DynamicFusion: Reconstruction and Tracking of Non-Rigid Scenes in Real-Time. In *Computer Vision and Pattern Recognition (CVPR)*.

Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Kim, Andrew J. Davison, Pushmeet Kohli, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. 2011.

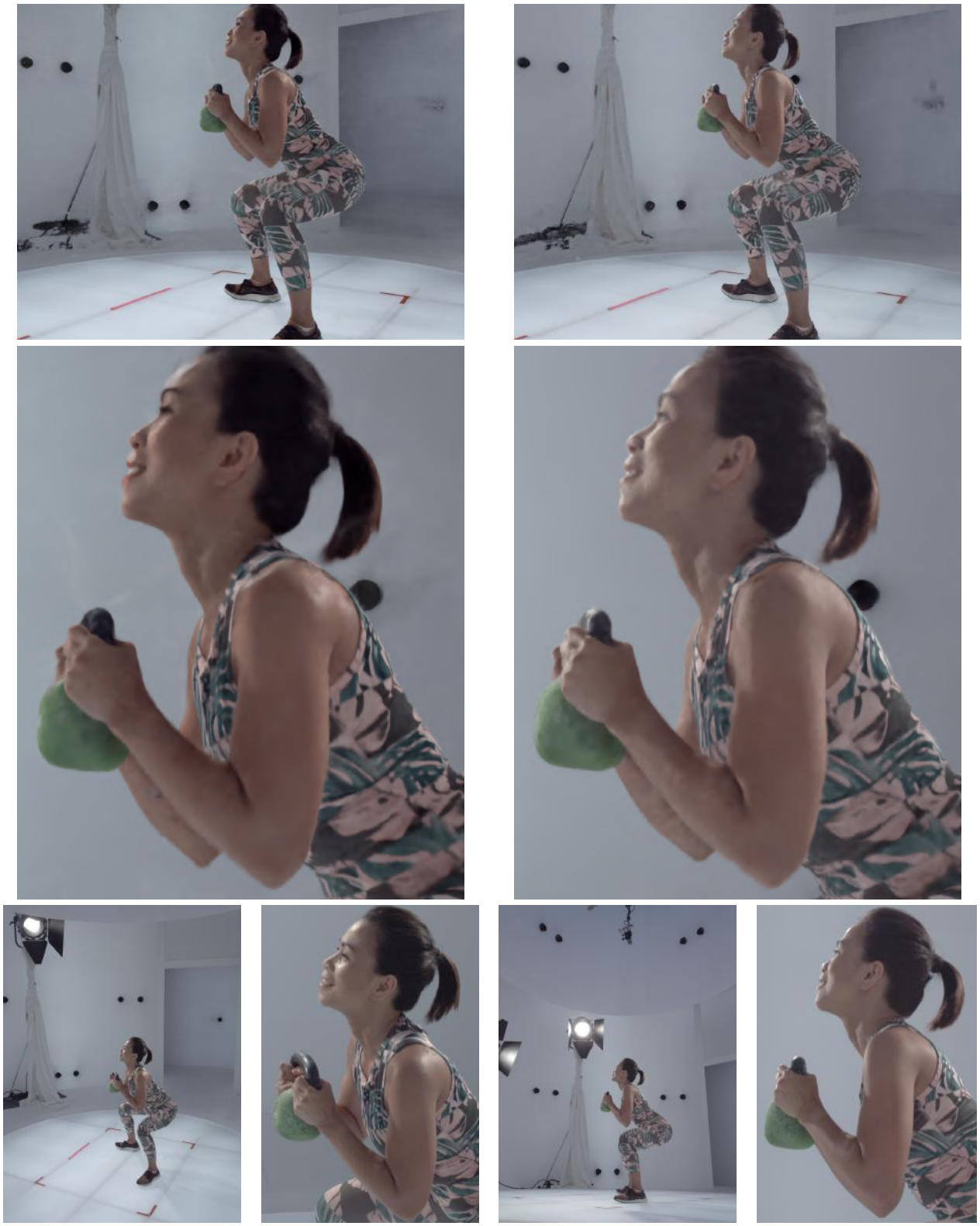


Fig. 12. We explore the upper quality bound of our proposed method using a highly controlled multi-view setting. We can extend our method such that it handles view-dependent effects. Results on the left are without view dependence, while those on the right are with view dependence. We show a full rendering by NR-NeRF (first row), zoom-ins thereof (second row), and input images from the two closest input cameras.

Fig. 13. The input (left) is reconstructed by NR-NeRF (middle). The bottom of the image exhibits local shadowing absent at other time steps, which leads to a high reconstruction error (right).

Fig. 14. The input (left) is reconstructed by NR-NeRF (middle) and rendered into a novel view (right). Due to strong regularization required by the problem setting, some scenes lead to compromises between sharpness and stability.

KinectFusion: Real-Time Dense Surface Mapping and Tracking. In *International Symposium on Mixed and Augmented Reality (ISMAR)*.

Dat Tien Ngo, Sanghyuk Park, Anne Jorstad, Alberto Crivellaro, Chang D. Yoo, and Pascal Fua. 2015. Dense Image Registration and Deformable Surface Reconstruction in Presence of Occlusions and Minimal Texture. In *International Conference on Computer Vision (ICCV)*.

Thu H Nguyen-Phuoc, Chuan Li, Stephen Balaban, and Yongliang Yang. 2018. RenderNet: A deep convolutional network for differentiable rendering from 3D shapes. In *Advances in Neural Information Processing Systems (NIPS)*.

Sergio Orts-Escolano, Christoph Rhemann, Sean Fanello, Wayne Chang, Adarsh Kowdle, Yury Degtyarev, David Kim, Philip L. Davidson, Sameh Khamis, Mingsong Dou, and et al. 2016. Holoportation: Virtual 3D Teleportation in Real-Time. In *Annual Symposium on User Interface Software and Technology*. 741–754.

M. R. Oswald, J. Stühmer, and D. Cremers. 2014. Generalized Connectivity Constraints for Spatio-temporal 3D Reconstruction. In *European Conference on Computer Vision (ECCV)*.

Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. 2019. Deepsdf: Learning continuous signed distance functions for shape representation. *International Conference on Computer Vision and Pattern Recognition (CVPR)* (2019).

Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. 2020. Deformable Neural Radiance Fields. *arXiv preprint arXiv:2011.12948* (2020).

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. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.). Curran Associates, Inc., 8024–8035. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

Sida Peng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. 2020. Neural Body: Implicit Neural Representations with Structured Latent Codes for Novel View Synthesis of Dynamic Humans. *arXiv e-prints* (2020). arXiv:2012.15838

Mathieu Perriollat, Richard Hartley, and Adrien Bartoli. 2011. Monocular Template-based Reconstruction of Inextensible Surfaces. *Int. J. Comput. Vision (IJCV)* 95, 2 (2011), 124–137.

Hanspeter Pfister, Matthias Zwicker, Jeroen van Baar, and Markus Gross. 2000. Surfels: Surface Elements as Rendering Primitives. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '00)*. ACM Press/Addison-Wesley Publishing Co., USA, 335–342. https://doi.org/10.1145/344779.344936

Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. 2020. D-NeRF: Neural Radiance Fields for Dynamic Scenes. *arxiv* (2020).

Gernot Riegler and Vladlen Koltun. 2020. Free View Synthesis. In *European Conference on Computer Vision (ECCV)*.

Kripasindhu Sarkar, Dushyant Mehta, Weipeng Xu, Vladislav Golyanik, and Christian Theobalt. 2020. Neural Re-Rendering of Humans from a Single Image. In *European Conference on Computer Vision (ECCV)*.

Johannes L Schonberger and Jan-Michael Frahm. 2016. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4104–4113.

Johannes Lutz Schönberger and Jan-Michael Frahm. 2016. Structure-from-Motion Revisited. In *Computer Vision and Pattern Recognition (CVPR)*.

Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. 2016. Pixelwise View Selection for Unstructured Multi-View Stereo. In *European Conference on Computer Vision (ECCV)*.

Jonathan Shade, Steven Gortler, Li-wei He, and Richard Szeliski. 1998. Layered Depth Images. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '98)*. Association for Computing Machinery, New York, NY, USA, 231–242. https://doi.org/10.1145/280814.280882

Aliaksandra Shysheya, Egor Zakharov, Kara-Ali Aliev, Renat Bashirov, Egor Burkov, Karim Iskakov, Aleksei Ivakhnenko, Yury Malkov, Igor Pasechnik, Dmitry Ulyanov, Alexander Vakhitov, and Victor Lempitsky. 2019. Textured Neural Avatars. In *Computer Vision and Pattern Recognition (CVPR)*.

Vikramjit Sidhu, Edgar Tretschk, Vladislav Golyanik, Antonio Agudo, and Christian Theobalt. 2020. Neural Dense Non-Rigid Structure from Motion with Latent Space Constraints. In *European Conference on Computer Vision (ECCV)*.

Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Niessner, Gordon Wetzstein, and Michael Zollhofer. 2019a. DeepVoxels: Learning Persistent 3D Feature Embeddings. In *Computer Vision and Pattern Recognition (CVPR)*.

Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. 2019b. Scene Representation Networks: Continuous 3D-Structure-Aware Neural Scene Representations. In *Advances in Neural Information Processing Systems*.

Miroslava Slavcheva, Maximilian Baust, Daniel Cremers, and Slobodan Ilic. 2017a. KillingFusion: Non-Rigid 3D Reconstruction Without Correspondences. In *Computer Vision and Pattern Recognition (CVPR)*.

Miroslava Slavcheva, Maximilian Baust, Daniel Cremers, and Slobodan Ilic. 2017b. Killingfusion: Non-rigid 3d reconstruction without correspondences. In *Computer Vision and Pattern Recognition (CVPR)*. 1386–1395.

Aljoscha Smolic, Karsten Mueller, Philipp Merkle, Christoph Fehn, Peter Kauff, Peter Eisert, and Thomas Wiegand. 2006. 3D video and free viewpoint video-technologies, applications and MPEG standards. In *2006 IEEE International Conference on Multimedia and Expo*. IEEE, 2161–2164.

Olga Sorkine and Marc Alexa. 2007. As-rigid-as-possible surface modeling. In *Symposium on Geometry Processing*, Vol. 4. 109–116.

Jonathan Starck, Gregor Miller, and Adrian Hilton. 2006. Volumetric Stereo with Silhouette and Feature Constraints. In *British Machine Vision Conference (BMVC)*.

Shufeng Tan and Michael L. Mayrovouniotis. 1995. Reducing data dimensionality through optimizing neural network inputs. *AIChE Journal* 41, 6 (1995), 1471–1480.

Yu Tao, Zerong Zheng, Kaiwen Guo, Jianhui Zhao, Dai Quionhai, Hao Li, Gerard Pons-Moll, and Yebin Liu. 2018. DoubleFusion: Real-time Capture of Human Performance with Inner Body Shape from a Depth Sensor. In *Computer Vision and Pattern Recognition (CVPR)*.

A. Tewari, O. Fried, J. Thies, V. Sitzmann, S. Lombardi, K. Sunkavalli, R. Martin-Brualla, T. Simon, J. Saragih, M. Nießner, R. Pandey, S. Fanello, G. Wetzstein, J.-Y. Zhu, C. Theobalt, M. Agrawala, E. Shechtman, D. B Goldman, and M. Zollhöfer. 2020. State of the Art on Neural Rendering. *Computer Graphics Forum (EG STAR 2020)* (2020).

Christian Theobalt, Naveed Ahmed, Hendrik Lensch, Marcus Magnor, and Hans-Peter Seidel. 2007. Seeing People in Different Light-Joint Shape, Motion, and Reflectance Capture. *IEEE Transactions on Visualization and Computer Graphics (TVCG)* 13, 4 (2007), 663–674.

Justus Thies, Michael Zollhöfer, and Matthias Nießner. 2019. Deferred neural rendering: image synthesis using neural textures. *ACM Transactions on Graphics* 38 (2019).

Edgar Tretschk, Ayush Tewari, Michael Zollhöfer, Vladislav Golyanik, and Christian Theobalt. 2020. DEMEA: Deep Mesh Autoencoders for Non-Rigidly Deforming Objects. In *European Conference on Computer Vision (ECCV)*.

Tony Tung, Shohei Nobuhara, and Takashi Matsuyama. 2009. Complete Multi-View Reconstruction of Dynamic Scenes from Probabilistic Fusion of Narrow and Wide Baseline Stereo. *International Conference on Computer Vision (ICCV)*, 1709–1716.

Ziyan Wang, Timur Bagautdinov, Stephen Lombardi, Tomas Simon, Jason Saragih, Jessica Hodgins, and Michael ZollhÃ¶fer. 2020. Learning Compositional Radiance Fields of Dynamic Human Heads. *arXiv e-prints* (2020).

Michael Waschbüsch, Stephan Würmlin, Daniel Cotting, Filip Sadlo, and Markus Gross. 2005. Scalable 3D video of dynamic scenes. *The Visual Computer* 21, 8 (2005), 629–638.

Chung-Yi Weng, Brian Curless, and Ira Kemelmacher-Shlizerman. 2020. Vid2Actor: Free-viewpoint Animatable Person Synthesis from Video in the Wild. *arXiv e-prints* (2020). arXiv:2012.12884

Wenqi Xian, Jia-Bin Huang, Johannes Kopf, and Changil Kim. 2020. Space-time Neural Irradiance Fields for Free-Viewpoint Video. *arXiv preprint* (2020).

Donglai Xiang, Hanbyul Joo, and Yaser Sheikh. 2019. Monocular total capture: Posing face, body, and hands in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 10965–10974.

Weipeng Xu, Avishek Chatterjee, Michael Zollhöfer, Helge Rhodin, Dushyant Mehta, Hans-Peter Seidel, and Christian Theobalt. 2018. MonoPerfCap: Human Performance Capture From Monocular Video. *ACM Trans. Graph.* 37, 2 (2018), 27:1–27:15.

Lin Yen-Chen. 2020. NeRF-pytorch. https://github.com/yenchenlin/nerf-pytorch/.

Rui Yu, Chris Russell, Neill D. F. Campbell, and Lourdes Agapito. 2015. Direct, Dense, and Deformable: Template-Based Non-Rigid 3D Reconstruction from RGB Video. In *International Conference on Computer Vision (ICCV)*.

T. Yu, K. Guo, F. Xu, Y. Dong, Z. Su, J. Zhao, J. Li, Q. Dai, and Y. Liu. 2017. BodyFusion: Real-Time Capture of Human Motion and Surface Geometry Using a Single Depth Camera. In *International Conference on Computer Vision (ICCV)*. 910–919.

Tao Yu, Zerong Zheng, Yuan Zhong, Jianhui Zhao, Qionghai Dai, Gerard Pons-Moll, and Yebin Liu. 2019. SimulCap : Single-View Human Performance Capture with Cloth Simulation. In *Conference on Computer Vision and Pattern Recognition(CVPR)*.

Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. 2020. NeRF++: Analyzing and Improving Neural Radiance Fields. *arXiv preprint arXiv:2010.07492* (2020).

L. Zhang, B. Curless, and S. M. Seitz. 2003. Spacetime stereo: shape recovery for dynamic scenes. In *Computer Vision and Pattern Recognition (CVPR)*.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *Computer Vision and Pattern Recognition (CVPR)*.

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

Hao Zhu, Hao Su, Peng Wang, Xun Cao, and Ruigang Yang. 2018. View Extrapolation of Human Body from a Single Image. In *Computer Vision and Pattern Recognition (CVPR)*.

Michael Zollhöfer, Matthias Nießner, Shahram Izadi, Christoph Rhemann, Christopher Zach, Matthew Fisher, Chenglei Wu, Andrew Fitzgibbon, Charles Loop, Christian Theobalt, and Marc Stamminger. 2014. Real-Time Non-Rigid Reconstruction Using an RGB-D Camera. *ACM Transactions on Graphics (TOG)* (2014).

Michael Zollhöfer, Patrick Stotko, Andreas Görlitz, Christian Theobalt, Matthias Nießner, Reinhard Klein, and Andreas Kolb. 2018. State of the Art on 3D Reconstruction with RGB-D Cameras. In *Computer graphics forum*, Vol. 37. Wiley Online Library, 625–652.