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

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

In this tech report, we present the current state of our ongoing work on reconstructing Neural Radiance Fields (NERF) of general non-rigid scenes via ray bending. Non-rigid NeRF (NR-NeRF) takes RGB images of a deforming object (*e.g.,* from a monocular video) as input and then learns a geometry and appearance representation that not only allows to reconstruct the input sequence but also to re-render any time step into novel camera views with high fidelity. In particular, we show that a consumer-grade camera is sufficient to synthesize convincing bullet-time videos of short and simple scenes. In addition, the resulting representation enables correspondence estimation across views and time, and provides rigidity scores for each point in the scene. We urge the reader to watch the supplemental videos for qualitative results. We will release our code.

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 with time. 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 rely on less constraints about the scene, compared to traditional approaches. The closest prior work to our method is NeRF [Mildenhall et al. 2020a], which builds a continuous implicit volume of the scene using multiple views from cameras with known extrinsic and intrinsic parameters. This implicit 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. We instead target the more challenging setting of using just a monocular video of the 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 multi-view reconstruction.

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 NeRF volume for geometry and appearance and (2) the scene deformations. The canonical volume is a static representation of the scene, 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 parameterized by a multi-layered perceptron (MLP) that takes point samples on the ray as well as a latent code for each image as input. Both the ray bending as well as the static scene MLPs are jointly trained on the monocular observations. Since the ray bending MLP deforms the entire space, independent of any camera parameters, we can render the deforming volume from a novel viewpoint after training. This formulation also allows us to compute dense correspondences (via the canonical scene volume) between all time steps.

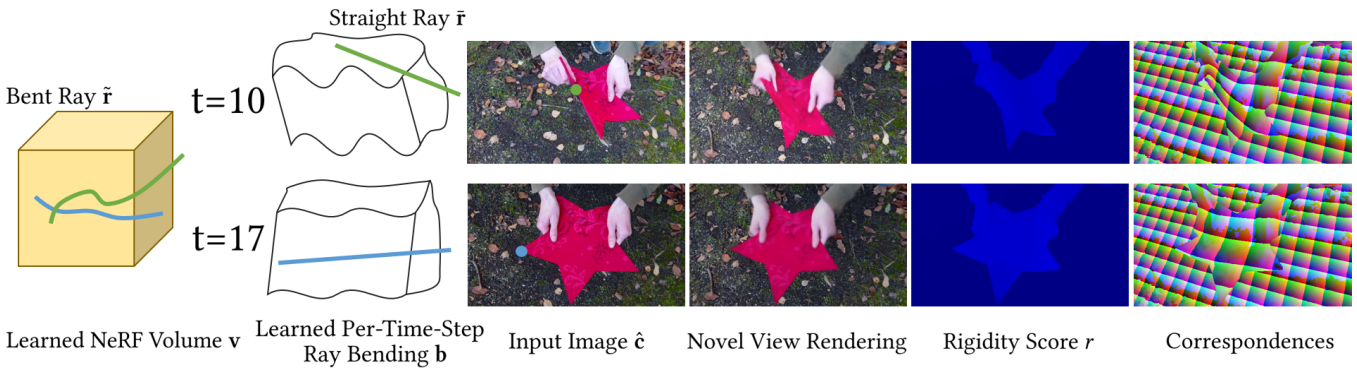

Fig. 1. Given only RGB input images $\{\hat{\mathrm{c}}_i\}_i$ of a deforming scene, *e.g.* from a monocular video, and the associated camera parameters $\{\mathrm{R}_i, \mathrm{t}_i, \mathrm{K}_i\}_i$, NR-NeRF accumulates geometry and appearance information from all input images in a single static NeRF volume v and models deformations as per-time-step space warpings b of that NeRF volume. The space warping is realized as a ray bending network that regresses an offset for any point in 3D space, conditioned on an auto-decoded per-time-step deformation latent code l_i. These offsets are applied to the straight input camera rays $\bar{\mathrm{r}}$ in order to bend them into the NeRF volume such that the input image is correctly reconstructed by integrating along the bent rays $\tilde{\mathrm{r}}$. In addition to enabling novel view synthesis, our learned representation provides rigidity scores r and correspondences via the NeRF volume.

The ray bending MLP disentangles the geometry of the scene from the scene deformations. This 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 regularizers as additional soft-constraints. A sparsity regularizer on the deformations encourages only sparse deformations of the volume, and thus helps to constrain the canonical volume. We further add a local-shape preserving regularizer that tries to preserve the local volume of the deforming scene by minimizing the divergence of the deformations. The sparsity regularizer is weighted such that it mainly encourages sparsity in the regions of the space that are visible and occupied. The hidden regions are constrained using the divergence regularizer. This is important for high quality results. Our results show high-fidelity reconstruction and novel view synthesis for a wide range of non-rigid scenes. We can also visualize the rigidity scores and dense correspondences. Fig. 1 contains an overview of our method.

To summarize, our main technical contributions are:

- A free viewpoint rendering method that only requires a monocular video of the deforming scene, enabled by the disentanglement of the scene into a static canonical volume and scene deformation.
- A rigidity network which can segment the scene into non-rigid foreground and rigid background without being directly supervised.
- Regularizers on the deformations which constrain the problem by encouraging small volume preserving deformations.

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

Earlier 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 incoming views [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 set-ups.

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 still 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 (NV) [Lombardi et al. 2019] learn object models which can be animated and rendered from novel views, given multi-view video data. NV encodes 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 re-render the scene from novel views or define an arbitrary observation trajectory of a virtual camera.

Concurrent Work as of November 2020. Concurrently to us, several methods have been proposed for similar tasks which are also yet to undergo peer-review. Unlike Pumarola et al. [2020], we demonstrate that our method can handle real scenes and supports background. Unlike Park et al. [2020], we show results on more challenging scenes, which are typically larger and include more motion, and we do not rely on pre-computed static background points for segmentation. Unlike Li et al. [2020], we do not rely on optical flow estimates or depth estimates for constraining the method. In contrast to Xian et al. [2020], our results are sharper in the rigid regions of the scene, less blurred in overall and less prone to halo-effects in the transitions from the background to the foreground. Furthermore, our method does not rely on video depth supervision and we believe that it can handle larger changes in the scene.

Note that we only use an auxiliary method for estimating the camera parameters. However, no reliance on pre-computed scene information apart from camera parameters, more challenging novel view scenarios, lower resolution training data, and significantly less compute make our results slightly less sharp than some concurrent work [Park et al. 2020].

3 METHOD

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

3.1 Background: Neural Radiance Fields

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

Let us consider a pixel (u, v) of an image $\hat{\mathrm{c}}_i$. In the case of a pinhole camera model, the associated ray $\mathrm{r}_{u,v}(j) = \mathrm{o} + j\mathrm{d}(u, v)$ can be calculated using $\mathrm{R}_i, \mathrm{t}_i$, and K_i, which yield the ray origin $\mathrm{o} \in \mathbb{R}^3$ and ray direction $\mathrm{d}(u, v) \in \mathbb{R}^3$. 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 visibility-weighted color c at (u, v):

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

where the visibility $V(j) = \exp(-\int_{j_n}^{j} o(\mathrm{r}_{u,v}(s))\, ds)$ accumulates the opacity along the ray from the near plane up to j.

In practice, the integrals are approximated by discrete points x along the ray. Ideally, ray points would cluster around regions with large $V(j) \cdot o(\mathrm{r}_{u,v}(j))$ along the ray since those regions contribute the most to the final color. However, since it is unknown beforehand where these regions are located, NeRF uses a coarse volume v_c with network weights θ_c and a fine volume 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 point samples along the ray. These coarse samples are used to estimate the visibility distribution V, from which fine samples are sampled. The fine volume is then evaluated at the combined set of coarse and fine ray points. We refer to the original paper [Mildenhall et al. 2020a] 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 $\mathrm{c} = \mathrm{c}(\mathrm{x})$. 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

To account for deformations, we use ray bending or space warping. For an image $\hat{\mathrm{c}}_i$, we want to render the canonical NeRF volume such that it reconstructs the input. To that end, we bend the straight rays sent out by the camera such that sampling and subsequently rendering the canonical NeRF along the bent rays yields $\hat{\mathrm{c}}_i$.

We implement ray bending as a ray bending network $\mathrm{b}(\mathrm{x}, \mathrm{l}_i) \in \mathbb{R}^3$ that regresses an offset at the ray point x under a deformation represented by l_i. The offset is then added to the ray point x before passing the result to the canonical NeRF volume, that is: $(\mathrm{c}, o) = \mathrm{v}(\mathrm{x} + \mathrm{b}(\mathrm{x}, \mathrm{l}_i))$. Note that v is not conditioned on l_i. We denote the bent version of the straight ray $\bar{\mathrm{r}}$ as $\tilde{\mathrm{r}}_{\mathrm{l}_i}(j) = \bar{\mathrm{r}}(j) + \mathrm{b}(\bar{\mathrm{r}}(j), \mathrm{l}_i)$.

Rigidity Network. However, we find that rigid parts of the scene are insufficiently constrained by this naive formulation. We reformulate $\mathrm{b}(\mathrm{x}, \mathrm{l}_i) \in \mathbb{R}^3$ as the combination of a raw offset $\mathrm{b}'(\mathrm{x}, \mathrm{l}_i)$ and a rigidity mask $r(\mathrm{x}) \in [0,1]$, *i.e.*, $\mathrm{b}(\mathrm{x}, \mathrm{l}_i) = r(\mathrm{x})\mathrm{b}'(\mathrm{x}, \mathrm{l}_i)$[1]. For rigid objects, we want to prevent deformations and hence desire $r(\mathrm{x}) = 0$, while for non-rigid objects, we want $r(\mathrm{x}) > 0$. This makes it easier for 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 r, which is jointly trained. Because the rigidity network is not conditioned on the latent code l_i, it is forced to share knowledge about rigidity 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 deformations.

3.3 Losses

Now that we have set up the architecture, let us see how we can optimize for all parameters $(\theta, \psi, \{\mathrm{l}_i\}_i)$ jointly. While the network weights are optimized as usual, the latent codes l_i are auto-decoded, *i.e.*, they are treated as free variables that are directly optimized for like network weights instead of getting 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 $\bar{\mathrm{r}}$ with coarse ray points $\bar{C} = \{\bar{\mathrm{r}}(j)\}_{j \in C}$ for a set C of uniformly sampled $j \in [j_n, j_f]$ and fine ray points $\bar{F} = \{\bar{\mathrm{r}}(j)\}_{j \in F}$ for a set F of importance sampled j. For a latent code l, the bent ray $\tilde{\mathrm{r}}_\mathrm{l}$ gives $\tilde{C} = \{\tilde{\mathrm{r}}_\mathrm{l}(j)\}_{j \in C}$ and $\tilde{F} = \{\tilde{\mathrm{r}}_\mathrm{l}(j)\}_{j \in F}$. 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:

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

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

Our problem setting is more underconstrained then NeRF's, and we find it necessary to regularize the bending of rays with further priors.

Offsets Loss. We regularize the offsets by a sparsity loss. Since we want air to be compressible and not hinder the optimization, we weigh the loss at each ray point by its occupancy. However, this would still apply a high weight to hidden ray points, which leads to severe artifacts when rendering novel views. We thus additionally weigh by visibility:

$$L_{offsets} = \frac{1}{|C|} \sum_{j \in C} w_j \cdot (\|\mathrm{b}'(\bar{\mathrm{r}}(j), \mathrm{l})\|_2 + \omega_{\text{rigidity}} r(\bar{\mathrm{r}}(j))), \qquad (3)$$

where we weigh each point by its visibility and occupancy $w_j = V(j) \cdot o(\tilde{\mathrm{r}}(j))$. We do not back-propagate into w_j. This loss has two advantages: (1) the gradient is independent of the magnitude of the offset, so unlike with an L2 loss, small and large offsets/motions are

[1] While using $r(\mathrm{x})$ ultimately encourages rigid points to remain static, using $r(\mathrm{x} + \mathrm{b}'(\mathrm{x}, \mathrm{l}_i))$ would discourage non-rigid points from entering rigid parts of the scene.

treated equally, and (2) relative to an L2 loss, it encourages sparsity in the offsets field, which fits our scenes. We found that applying the offsets loss to the masked offsets works worse than applying it to the regressed rigidity mask and raw offsets separately.

Divergence Loss. Since the offsets loss only constraints 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:

$$L_{divergence} = \frac{1}{|C|} \sum_{j \in C} w'_j \cdot |\mathrm{div}(\mathrm{b}(\bar{\mathrm{r}}(j), \mathrm{l}))|, \tag{4}$$

where $w'_j = o(\tilde{\mathrm{r}}(j))$, which we do not back-propagate into, and the divergence div is the divergence of b with respect to the position $\bar{\mathrm{r}}(j)$.

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

$$L = L_{\text{data}} + \omega_{\text{offsets}} L_{\text{offsets}} + \omega_{\text{divergence}} L_{\text{divergence}}, \tag{5}$$

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.

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 of NeRF. 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: slower and exact, or faster and approximate.

Exact: We obtain the direction of the bent ray $\tilde{\mathrm{r}}$ at a point $\tilde{\mathrm{r}}(j)$ via the chain rule as $\nabla_j \tilde{\mathrm{r}}(j) = \frac{\partial \tilde{\mathrm{r}}(j)}{\partial \bar{\mathrm{r}}(j)} \cdot \frac{\partial \bar{\mathrm{r}}(j)}{\partial j} = J \cdot \mathrm{d}$, where J is the 3×3 Jacobian and 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 $\tilde{\mathrm{r}}(j)$ and the previous point $\tilde{\mathrm{r}}(j-1)$ along the bent ray (which is closer to the camera).

3.4 Training Details

We initialize $\{\mathrm{l}_i\}_i$ to zero vectors. We use the same architecture for the NeRF volume as in the original paper [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. Scenes usually consist of 80 to 300 images, at resolutions of 480×270 (Blackmagic, Sony XZ2) or 512×384 (Kinect Azure). 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. We have found the following ranges to be sufficient for a wide range of scenarios: ω_{rigidity} lies in $[0.1, 10]$ and typically is 1, ω_{offsets} lies in $[0.01, 1]$ and typically is 0.06 or 0.6, and $\omega_{\text{divergence}}$ lies in $[0.1, 10]$ and typically is 1 or 3. We start the training with each weight set to $\frac{1}{100}$th of its value, and then exponentially increase it until its reaches its full value at the end of training.

3.5 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].

If the camera extrinsics and intrinsics are not given, we estimate them using Structure-from-Motion implemented by COLMAP [Schönberger and Frahm 2016; Schönberger et al. 2016]. We find COLMAP to be robust to non-rigid "outliers".

We use FFJORD's [Grathwohl et al. 2018] fast, approximate, unbiased divergence estimation, which reduces the computational costs of the divergence estimation compared to an exact computation by a factor of three in our case.

As we want smooth deformations, we only use positional encoding for the input to the canonical NeRF volume, not for the input to the ray bending network.

4 RESULTS

4.1 Data

We show results on 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. Appendix A contains a list of tips for recording scenes for NR-NeRF.

4.2 Qualitative Results

We present qualitative results that visualize the scene from both the training input views as well as novel views. In addition, we visualize the rigidity scores and correspondences, and show how the rigidity scores can be used to enforce a stable background. Furthermore, we show results of a model that takes view-dependent

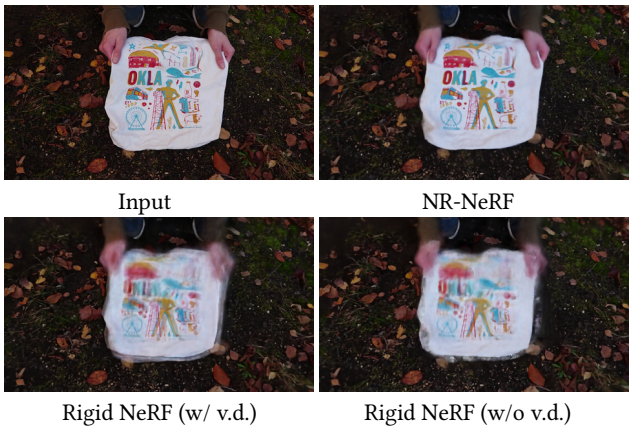

Fig. 2. Rigid NeRF on non-rigid input data, with and without view dependence (v.d.).

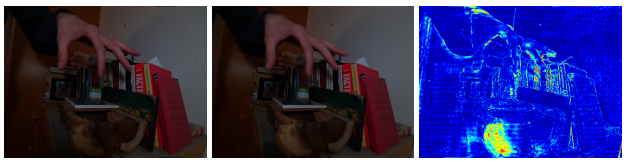

Fig. 3. The input (left) is reconstructed by NR-NeRF (middle). The bottom of the image exhibits local shadowing absent at other time steps, which leads to a high reconstruction error (right).

effects into account. For many more qualitative results, we refer to the supplemental video.

Comparison to Rigid NeRF. We compare our method to rigid NeRF [Mildenhall et al. 2020a]. As Fig. 2 shows, not accounting for deformations leads to blur. Interestingly, 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.

Error Maps. We visualize the reconstruction error of the input images. We first compute the norm of the RGB difference (RGB values in $[0, 1]^3$), then normalize them by the maximum difference, $\sqrt{3 \cdot 1^2}$, multiply by 10 to exaggerate the error for better visualization, and finally apply the jet color scheme. Thus, any pixel error above 10% is shown as saturated red. Fig. 3 contains an example.

Novel View Synthesis. We can re-render the learned representation into any novel view at any time step, as shown in Fig. 4. We refer to the supplemental video for numerous additional results.

Rigidity. 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 $\sum_{k=0}^{j-1} w_k$ 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. Fig. 1 shows an example.

Correspondences. We can use our reconstructions to estimate consistent correspondences across different camera views and time steps. To visualize this, we treat the canonical NeRF 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 model 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. Fig. 1 shows an example.

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. To enforce a stable background at test time, we can 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.

View Dependence. We can extend NR-NeRF to optionally model view-dependent effects. However, as shown in the supplemental video, we find our formulation to lead to hazy artifacts. We hypothesize that the combination of both significant motion and novel views significantly different from input views to be too underconstrained for view-dependent effects. For example, similar to the rigid NeRF results presented earlier, non-rigid NeRF might exploit 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.

5 LIMITATIONS AND FUTURE WORK

For simplicity, the discrete integration along the bent ray for integration uses interval lengths given by the straight ray, not the bent 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 networks.

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 differently 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, 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.

Fig. 4. The input (left) is reconstructed by NR-NeRF (middle) and re-rendered into a novel view (right).

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. 3 demonstrates.

Rendering parts of the scene barely or not at all observed in the training data would not lead to realistic results. Further assumptions on the types of scenes are listed in Appendix A in the form of practical tips for recording new scenes.

6 CONCLUSION

We presented a method for free viewpoint rendering of a deforming scene just using a monocular video as input. This is done by disentangling the dynamic scene into a canonical scene volume and scene deformation. We have shown that space warping in the form of ray bending is a promising scene deformation model for volumetric representations like NeRF. Both the static scene and scene deformation are jointly learned. At test time, the scene can be rendered from a new viewpoint using volumetric integration. Furthermore, we have demonstrated that background instability, a problem also noted by concurrent work [Park et al. 2020], can be mitigated in an unsupervised fashion by learning a rigidity mask for the offsets field. We further introduced sparsity and volume preserving regularizers to constrain the problem. We demonstrated several high-quality reconstruction and novel view synthesis results of general dynamic scenes, as well as dense correspondences between the reconstructions of different images. While several limitations exist as explained in Sec. 5, we hope that the technical contributions introduced will inspire future work.

Acknowledgements. This work was supported by the ERC Consolidator Grant 4DReply (770784), and an Oculus research grant.

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.

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).

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).

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.

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.

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. 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

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)*.
Ayush Tewari, Ohad Fried, Justus Thies, Vincent Sitzmann, Stephen Lombardi, Kalyan Sunkavalli, Ricardo Martin-Brualla, Tomas Simon, Jason Saragih, Matthias Nießner, et al. 2020. State of the Art on Neural Rendering. *arXiv preprint arXiv:2004.03805* (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.
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.
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)*.
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.

A PRACTICAL TIPS

When recording a scene for use with our method, please take the following tips into account:

- Avoid blur (*e.g.*, motion blur or out-of-focus blur).
- Keep camera settings like color temperature and focal length fixed.
- Avoid lens distortions or estimate distortion parameters for undistortion.
- Stick to front-facing camera paths that capture most of the scene in all images.
- Use sufficient lighting and avoid changing it while recording.
- Avoid self-shadowing.
- Only record Lambertian surfaces, avoid view-dependent effects like specularities.
- The background needs to be static and dominant enough for SfM to estimate extrinsics.
- Limited scene size: Ensure that the background is not more than an order of magnitude further from the camera compared to the non-rigid foreground.