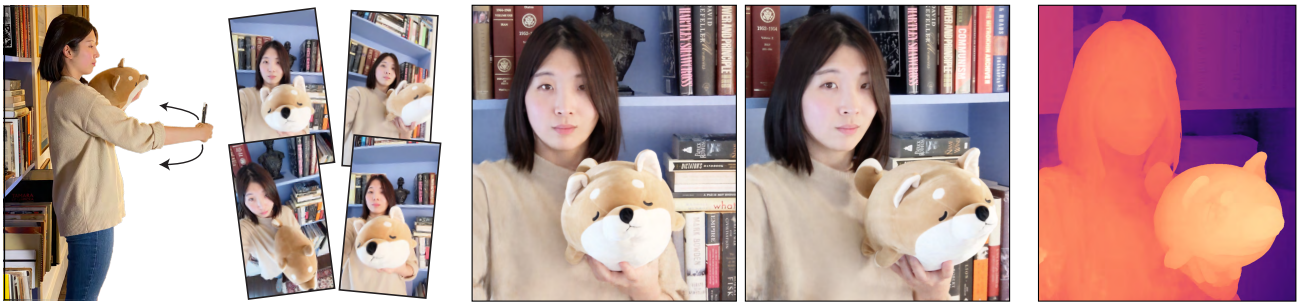
Nerfies: Deformable Neural Radiance Fields

Keunhong Park[1*] Utkarsh Sinha[2] Jonathan T. Barron[2] Sofien Bouaziz[2]
Dan B Goldman[2] Steven M. Seitz[1,2] Ricardo Martin-Brualla[2]

[1]University of Washington [2]Google Research
nerfies.github.io



(a) casual capture (b) input images (c) nerfie novel views (d) novel view depth

Figure 1: We reconstruct photo-realistic *nerfies* from a user casually waving a mobile phone (a). Our system uses selfie photos/videos (b) to produce a free-viewpoint representation with accurate renders (c) and geometry (d). Please see video.

Abstract

We present the first method capable of photorealistically reconstructing deformable scenes using photos/videos captured casually from mobile phones. Our approach augments neural radiance fields (NeRF) by optimizing an additional continuous volumetric deformation field that warps each observed point into a canonical 5D NeRF. We observe that these NeRF-like deformation fields are prone to local minima, and propose a coarse-to-fine optimization method for coordinate-based models that allows for more robust optimization. By adapting principles from geometry processing and physical simulation to NeRF-like models, we propose an elastic regularization of the deformation field that further improves robustness. We show that our method can turn casually captured selfie photos/videos into deformable NeRF models that allow for photorealistic renderings of the subject from arbitrary viewpoints, which we dub "nerfies." We evaluate our method by collecting time-synchronized data using a rig with two mobile phones, yielding train/validation images of the same pose at different viewpoints. We show that our method faithfully reconstructs non-rigidly deforming scenes and reproduces unseen views with high fidelity.

1. Introduction

High quality 3D human scanning has come a long way – but the best results currently require a specialized lab with many synchronized lights and cameras, e.g., [14, 15, 18]. What if you could capture a photorealistic model of yourself (or someone else) just by waving your mobile phone camera? Such a capability would dramatically increase accessibility and applications of 3D modeling technology.

Modeling people with hand-held cameras is especially challenging due both to 1) nonrigidity – our inability to stay perfectly still, and 2) challenging materials like hair, glasses, and earrings that violate assumptions used in most reconstruction methods. In this paper we introduce an approach to address both of these challenges, by generalizing Neural Radiance Fields (NeRF) [32] to model shape deformations. Our technique recovers high fidelity 3D reconstructions from short videos, providing free-viewpoint visualizations while accurately capturing hair, glasses, and other complex, view-dependent materials, as shown in Figure 1. A special case of particular interest is capturing a 3D self-portrait – we call such casual 3D selfie reconstructions *nerfies*.

Rather than represent shape explicitly, NeRF [32] uses a neural network to encode color and density as a function of location and viewing angle, and generates novel views using volume rendering. Their approach produces 3D visualizations of unprecedented quality, faithfully representing thin

*Work done while the author was an intern at Google.

structures, semi-transparent materials, and view-dependent effects. To model non-rigidly deforming scenes, we generalize NeRF by introducing an additional component: A canonical NeRF model serves as a template for all the observations, supplemented by a deformation field for each observation that warps 3D points in the frame of reference of an observation into the frame of reference of the canonical model. We represent this deformation field as a multi-layer perceptron (MLP), similar to the radiance field in NeRF. This deformation field is conditioned on a per-image learned latent code, allowing it to vary between observations.

Without constraints, the deformation fields are prone to distortions and over-fitting. We employ a similar approach to the elastic energy formulations that have seen success for mesh fitting [7, 12, 45, 46]. However, our volumetric deformation field formulation greatly simplifies such regularization, because we can easily compute the Jacobian of the deformation field through automatic differentiation, and directly regularize its singular values.

To robustly optimize the deformation field, we propose a novel coarse-to-fine optimization scheme that modulates the components of the input positional encoding of the deformation field network by frequency. By zeroing out the high frequencies at the start of optimization, the network is limited to learn smooth deformations, which are later refined as higher frequencies are introduced into the optimization.

For evaluation, we capture image sequences from a rig of two synchronized, rigidly attached, calibrated cameras, and use the reconstruction from one camera to predict views from the other. We plan to release the code and data.

In summary, our contributions are: ① an extension to NeRF to handle non-rigidly deforming objects that optimizes a deformation field per observation; ② rigidity priors suitable for deformation fields defined by neural networks; ③ a coarse-to-fine regularization approach that modulates the capacity of the deformation field to model high frequencies during optimization; ④ a system to reconstruct free-viewpoint selfies from casual mobile phone captures.

2. Related Work

Non-Rigid Reconstruction: Non-rigid reconstruction decomposes a scene into a geometric model and a deformation model that deforms the geometric model for each observation. Earlier works focused on sparse representations such as keypoints projected onto 2D images [10, 48], making the problem highly ambiguous. Multi-view captures [14, 15] simplify the problem to one of registering and fusing 3D scans [22]. DynamicFusion [33] uses a single RGBD camera moving in space, solving jointly for a canonical model, a deformation, and camera pose. More recently, learning-based methods have been used to find correspondences useful for non-rigid reconstruction [9, 39]. Unlike prior work, our method does not require depth nor multi-view capture systems and works on monocular RGB inputs. Most similar to our work, Neural Volumes [25] learns a 3D representation of a deformable scene using a voxel grid and warp field regressed from a 3D CNN. However, their method requires dozens of synchronized cameras and our evaluation shows that it does not extend to sequences captured from a single camera. Yoon *et al*. [52] reconstruct dynamic scenes from moving camera trajectories, but their method relies on strong semantic priors, in the form of monocular depth estimation, which are combined with multi-view cues. OFlow [34] solves for temporal flow-fields using ODEs, and thus requires temporal information. ShapeFlow [20] learns 3D shapes a divergence-free deformations of a learned template. Instead, we propose an elastic energy regularization.

Domain-Specific Modeling: Many reconstruction methods use domain-specific knowledge to model the shape and appearance of categories with limited topological variation, such as faces [4, 6, 8], human bodies [26, 51], and animals [11, 57]. Although some methods show impressive results in monocular face reconstruction from color and RGBD cameras [56], such models often lack detail (e.g., hair), or do not model certain aspects of a category (e.g., eyewear or garments). Recently, image translation networks have been applied to improve the realism of composited facial edits [16, 21]. In contrast, our work does not rely on domain-specific knowledge, enabling us to model the whole scene, including eyeglasses and hair for human subjects.

Coordinate-based Models: Our method builds on the recent success of coordinate-based models, which encode a spatial field in the weights of a multilayer perceptron (MLP) and require significantly less memory compared to discrete representations. These methods have been used to represent shapes [13, 31, 35] and scenes [32, 44]. Of particular interest are NeRFs [32], that use periodic positional encoding layers [43, 47] to increase resolution, and whose formulation has be extended to handle different lighting conditions [3, 29], transient objects [29], large scenes [24, 53] and to model object categories [41]. Our work extends NeRFs to handle non-rigid scenes.

Concurrent Work: Two concurrent works [37, 49] propose to represent deformable scenes using a translation field in conjunction with a template. This is similar to our framework with the following differences: ① we condition the deformation with a per-example latent [5] instead of time [37]; ② propose an as-rigid-as-possible regularization of the deformation field while NR-NeRF [49] penalizes the divergence of the translation field; ③ propose a coarse-to-fine regularization to prevent getting stuck in local minima; and ④ propose an improved SE(3) parameterization of the deformation field. Other concurrent works [23, 50] reconstruct space-time videos by recovering time-varying NeRFs while leveraging external supervision such as monocular depth estimation and flow-estimation to resolve ambiguities.

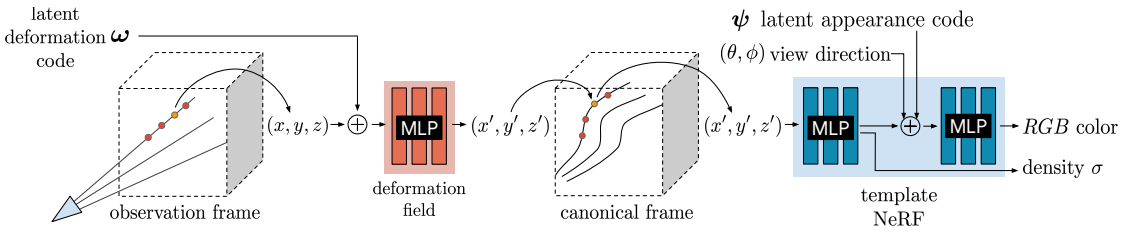


Figure 2: We associate a latent deformation code ($\boldsymbol{\omega}$) and an appearance code ($\boldsymbol{\psi}$) to each image. We trace the camera rays in the observation frame and transform samples along the ray to the canonical frame using a deformation field encoded as an MLP that is conditioned on the deformation code $\boldsymbol{\omega}$. We query the template NeRF [32] using the transformed sample (x', y', z'), viewing direction (θ, ϕ) and appearance code $\boldsymbol{\psi}$ as inputs to the MLP and integrate samples along the ray following NeRF.

3. Deformable Neural Radiance Fields

Here we describe our method for modeling non-rigidly deforming scenes given a set of casually captured images of the scene. We decompose a non-rigidly deforming scene into a template volume represented as a neural radiance field (NeRF) [32] (§3.1) and a per-observation deformation field (§3.2) that associates a point in observation coordinates to a point on the template (overview in Fig. 2). The deformation field is our key extension to NeRF and allows us to represent moving subjects. Jointly optimizing a NeRF together with a deformation field leads to an under-constrained optimization problem. We therefore introduce an elastic regularization on the deformation (§3.3), a background regularization (§3.4), and a continuous, coarse-to-fine annealing technique that avoids bad local minima (§3.5).

3.1. Neural Radiance Fields

A neural radiance field (NeRF) is a continuous, volumetric representation. It is a function $F : (\mathbf{x}, \mathbf{d}, \boldsymbol{\psi}_i) \rightarrow (\mathbf{c}, \sigma)$ which maps a 3D position $\mathbf{x} = (x, y, z)$ and viewing direction $\mathbf{d} = (\phi, \theta)$ to a color $\mathbf{c} = (r, g, b)$ and density σ. In practice, NeRF maps the inputs $\mathbf{x}$ and $\mathbf{d}$ using a sinusoidal positional encoding $\gamma : \mathbb{R}^3 \rightarrow \mathbb{R}^{3+6m}$ defined as $\gamma(\mathbf{x}) = \big(\mathbf{x}, \cdots, \sin\left(2^k \pi \mathbf{x}\right), \cos\left(2^k \pi \mathbf{x}\right), \cdots\big)$, where m is a hyper-parameter that controls the total number of frequency bands

and $k \in \{0, \ldots, m-1\}$. This function projects a coordinate vector $\mathbf{x} \in \mathbb{R}^3$ to a high dimensional space using a set of sine and cosine functions of increasing frequencies. This allows the MLP to model high-frequency signals in low-frequency domains as shown in [47]. Coupled with volume rendering techniques, NeRFs can represent scenes with photo-realistic quality. We build upon NeRF to tackle the problem of capturing deformable scenes.

Similar to NeRF-W [29], we also provide an appearance latent code $\boldsymbol{\psi}_i$ for each observed frame $i \in \{1, \ldots, n\}$ that modulates the color output to handle appearance variations between input frames, e.g., exposure and white balance.

The NeRF training procedure relies on the fact that given a 3D scene, two intersecting rays from two different cameras should yield the same color. Disregarding specular reflection and transmission, this assumption is true for all static scenes. Unfortunately, many scenes are not completely static; e.g., it is hard for people to stay completely still when posing for a photo, or worse, when waving a phone when capturing themselves in a selfie video.

3.2. Neural Deformation Fields

With the understanding of this limitation, we extend NeRF to allow the reconstruction of non-rigidly deforming scenes. Instead of directly casting rays through a NeRF, we use it as a canonical template of the scene. This template contains the relative structure and appearance of the scene while a rendering will use a non-rigidly deformed version of the template (see Fig. 3 for an example). DynamicFusion [33] and Neural Volumes [25] also model a template and a per-frame deformation, but the deformation is defined on mesh points and on a voxel grid respectively, whereas we model it as a continuous function using an MLP.

We employ an observation-to-canonical deformation for every frame $i \in \{1, \ldots, n\}$, where n is the number of observed frames. This defines a mapping $T_i : \mathbf{x} \rightarrow \mathbf{x}'$ that maps all observation-space coordinates $\mathbf{x}$ to a canonical-space coordinate $\mathbf{x}'$. We model the deformation fields for all time steps using a mapping $T : (\mathbf{x}, \boldsymbol{\omega}_i) \rightarrow \mathbf{x}'$, which is conditioned on a per-frame learned latent deformation code $\boldsymbol{\omega}_i$. Each latent code encodes the state of the scene in frame i. Given a canonical-space radiance field F and a observation-to-canonical mapping T, the observation-space radiance field can be evaluated as:

$$G(\mathbf{x}, \mathbf{d}, \boldsymbol{\psi}_i, \boldsymbol{\omega}_i) = F\left(T(\mathbf{x}, \boldsymbol{\omega}_i), \mathbf{d}, \boldsymbol{\psi}_i\right) . \qquad (1)$$

When rendering, we simply cast rays and sample points in the observation frame and then use the deformation field to map the sampled points to the template, see Fig. 2.

A simple model of deformation is a displacement field $V : (\mathbf{x}, \boldsymbol{\omega}_i) \rightarrow \mathbf{t}$, defining the transformation as $T(\mathbf{x}, \boldsymbol{\omega}_i) = \mathbf{x} + V(\mathbf{x}, \boldsymbol{\omega}_i)$. This formulation is sufficient to represent all continuous deformations; however, rotating a group of points with a translation field requires a different translation for each point, making it difficult to rotate regions of the scene

observation frame canonical frame

Figure 3: Visualizations of the recovered 3D model in the observation and canonical frames of reference, with insets showing orthographic views in the forward and left directions. Note the right-to-left and front-to-back displacements between the observation and canonical model, which are modeled by the deformation field for this observation.

simultaneously. We therefore formulate the deformation using a dense SE(3) field $W : (\mathbf{x}, \boldsymbol{\omega}_i) \rightarrow \mathrm{SE}(3)$. An SE(3) transform encodes rigid motion, allowing us to rotate a set of distant points with the same parameters.

We encode a rigid transform as a screw axis [28] $\mathcal{S} = (\mathbf{r}; \mathbf{v}) \in \mathbb{R}^6$. Note that $\mathbf{r} \in \mathfrak{so}(3)$ encodes a rotation where $\hat{\mathbf{r}} = \mathbf{r}/\|\mathbf{r}\|$ is the axis of rotation and $\theta = \|\mathbf{r}\|$ is the angle of rotation. The exponential of $\mathbf{r}$ (also known as Rodrigues' formula [38]) yields a rotation matrix $e^{\mathbf{r}} \in \mathrm{SO}(3)$:

$$e^{\mathbf{r}} \equiv e^{[\mathbf{r}]_\times} = \mathbf{I} + \frac{\sin\theta}{\theta}[\mathbf{r}]_\times + \frac{1-\cos\theta}{\theta^2}[\mathbf{r}]_\times^2 , \qquad (2)$$

where $[\mathbf{x}]_\times$ denotes the cross-product matrix of a vector $\mathbf{x}$.

Similarly, the translation encoded by the screw motion $\mathcal{S}$ can be recovered as $\mathbf{p} = \mathbf{G}\mathbf{v}$ where

$$\mathbf{G} = \mathbf{I} + \frac{1-\cos\theta}{\theta^2}[\mathbf{r}]_\times + \frac{\theta-\sin\theta}{\theta^3}[\mathbf{r}]_\times^2 . \qquad (3)$$

Combining these formulas and using the exponential map, we get the transformed point as $\mathbf{x}' = e^{\mathcal{S}}\mathbf{x} = e^{\mathbf{r}}\mathbf{x} + \mathbf{p}$.

As mentioned before, we encode the transformation field in an MLP $W : (\mathbf{x}, \boldsymbol{\omega}_i) \rightarrow (\mathbf{r}, \mathbf{v})$ using a NeRF-like architecture, and represent the transformation of every frame i by conditioning on a latent code $\boldsymbol{\omega}_i$. We optimize the latent code through an embedding layer [5]. Like with the template, we map the input $\mathbf{x}$ using positional encoding γ_α (see §3.5). An important property of the $\mathfrak{se}(3)$ representation is that $e^{\mathcal{S}}$ is the identity when $\mathcal{S} = \mathbf{0}$. We therefore initialize the weights of the last layer of the MLP from $\mathcal{U}(-10^{-5}, 10^{-5})$ to initialize the deformation near the identity.

3.3. Elastic Regularization

The deformation field adds ambiguities that make optimization more challenging. For example, an object moving backwards is visually equivalent to it shrinking in size, with many solutions in between. These ambiguities lead to under-constrained optimization problems which yield implausible results and artifacts (see Fig. 6). It is therefore crucial to introduce priors that lead to a more plausible solution.

example inputs ground truth elastic off elastic on

Figure 4: Our elastic regularization helps when the scene is under-constrained. This capture only contains 20 input images with the cameras biased towards one side of the face resulting in an under constrained problem. Elastic regularization helps resolve the ambiguity and leads to less distortion.

It is common in geometry processing and physics simulation to model non-rigid deformations using elastic energies measuring the deviation of local deformations from a rigid motion [7, 12, 45, 46]. In the vision community, these energies have been extensively used for the reconstruction and tracking of non-rigid scenes and objects [15, 33, 55] making them good candidates for our approach. While they have been most commonly used for discretized surfaces, e.g., meshes, we can apply a similar concept in the context of our continuous deformation field.

Elastic Energy: For a fixed latent code $\boldsymbol{\omega}_i$, our deformation field T is a non-linear mapping from observation-coordinates in $\mathbb{R}^3$ to canonical coordinates in $\mathbb{R}^3$. The Jacobian $\mathbf{J}_T(\mathbf{x})$ of this mapping at a point $\mathbf{x} \in \mathbb{R}^3$ describes the best linear approximation of the transformation at that point. We can therefore control the local behavior of the deformation through $\mathbf{J}_T$ [42]. Note that unlike other approaches using discretized surfaces, our continuous formulation allows us to directly compute $\mathbf{J}_T$ through automatic differentiation of the MLP. There are several ways to penalize the deviation of the Jacobian $\mathbf{J}_T$ from a rigid transformation. Considering the singular-value decomposition of the Jacobian $\mathbf{J}_T = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$, multiple approaches [7, 12] penalize the deviation from the closest rotation as $\|\mathbf{J}_T - \mathbf{R}\|_F^2$, where $\mathbf{R} = \mathbf{V}\mathbf{U}^T$ and $\|\cdot\|_F$ is the Frobenius norm. We opt to directly work with the singular values of $\mathbf{J}_T$ and measure its deviation from the identity. The log of the singular values gives equal weight to a contraction and expansion of the same factor, and we found it to perform better. We therefore penalize the deviation of the log singular values from zero:

$$L_{\text{elastic}}(\mathbf{x}) = \|\log\boldsymbol{\Sigma} - \log\mathbf{I}\|_F^2 = \|\log\boldsymbol{\Sigma}\|_F^2 , \qquad (4)$$

where log here is the matrix logarithm.

Robustness: Although humans are mostly rigid, there are some movements which can break our assumption of local rigidity, e.g., facial expressions which locally stretch and compress our skin. We therefore remap the elastic energy

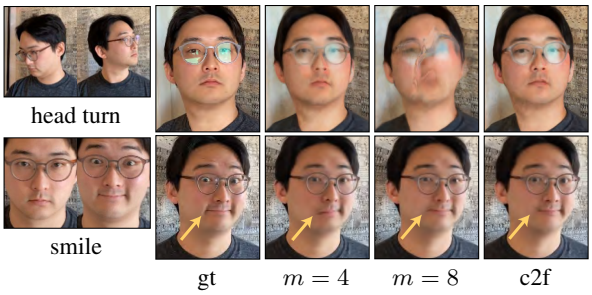


Figure 5: In this capture, the subject rotates their head (top) and smiles (bottom). With $m = 4$ positional encoding frequencies, the deformation model does not capture the smile, while it fails to rotate the head with $m = 8$ frequencies. With coarse-to-fine regularization (c2f) the model captures both.

defined above using a robust loss:

$$L_{\text{elastic-r}}(\mathbf{x}) = \rho\left(\|\log \mathbf{\Sigma}\|_F, c\right), \tag{5}$$

$$\rho(x, c) = \frac{2(x/c)^2}{(x/c)^2 + 4}. \tag{6}$$

where $\rho(\cdot)$ is the Geman-McClure robust error function [17] parameterized with hyperparameter $c = 0.03$ as per Barron [2]. This robust error function causes the gradients of the loss to fall off to zero for large values of the argument, thereby reducing the influence of outliers during training.

Weighting: We allow the deformation field to behave freely in empty space, since the subject moving relative to the background requires a non-rigid deformation somewhere in space. We therefore weight the elastic penalty at each sample along the ray by its contribution to the rendered view, *i.e.* w_i in Eqn. 5 of NeRF [32].

3.4. Background Regularization

The deformation field is unconstrained and therefore everything is free to move around. We optionally add a regularization term which prevents the background from moving. Given a set of 3D points in the scene which we know should be static, we can penalize any deformations at these points. For example, camera registration using structure from motion produces a set of 3D feature points that behave rigidly across at least some set of observations. Given these static 3D points $\{\mathbf{x}_1 \ldots, \mathbf{x}_K\}$, we penalize movement as:

$$L_{\text{bg}} = \frac{1}{K}\sum_{k=1}^{K}\left\|T(\mathbf{x}_k) - \mathbf{x}_k\right\|_2. \tag{7}$$

In addition to keeping the background points from moving, this regularization also has the benefit of aligning the observation coordinate frame to the canonical coordinate frame.

3.5. Coarse-to-Fine Deformation Regularization

A common trade-off that arises during registration and flow estimation is the choice between modeling minute versus large motions, that can lead to overly smooth results or

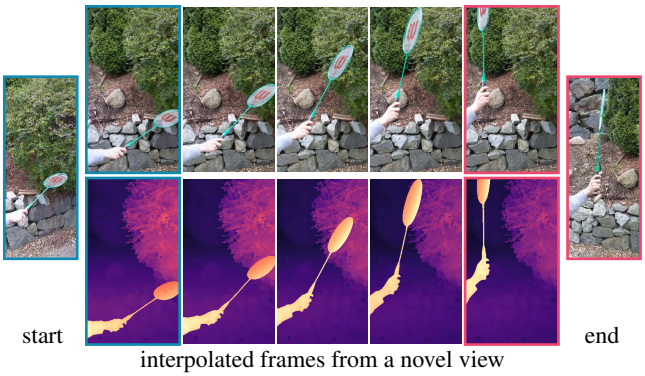


Figure 6: Novel views synthesized by linearly interpolating the deformation latent codes of two frames of the BADMINTON (left and right) show a smooth racquet motion.

incorrect registration (local minima). Coarse-to-fine strategies circumvent the issue by first solving the problem in low-resolution, where motion is small, and iteratively upscaling the solution and refining it [27]. We observe that our deformation model suffers from similar issues, and propose a coarse-to-fine regularization to mitigate them.

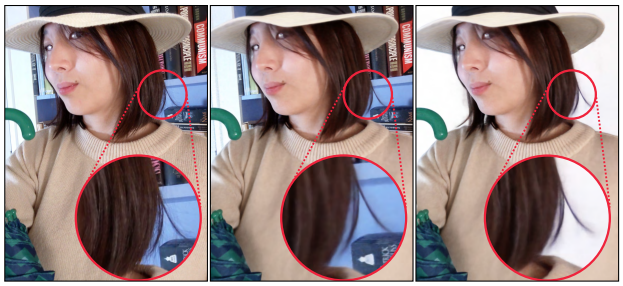
Recall the positional encoding parameter m introduced in §3.1 that controls the number of frequency bands used in the encoding. Tancik *et al*. [47] show that controls it the smoothness of the network: a low value of m results in a low-frequency bias (low resolution) while a higher value of m results in a higher-frequency bias (high resolution).

Consider a motion like in Fig. 5, where subject rotates their head and smiles. With a small m for the deformation field, the model cannot capture the minute motion of the smile; conversely, with a larger m, the model fails to correctly rotate the head because the template overfits to an underoptimized deformation field. To overcome this trade-off, we propose a coarse-to-fine approach that starts with a low-frequency bias and ends with a high-frequency bias.

Tancik *et al*. [47] show that positional encoding can be interpreted in terms of the Neural Tangent Kernel (NTK) [19] of NeRF's MLP: a stationary interpolating kernel where m controls a tunable "bandwidth" of that kernel. A small number of frequencies induces a wide kernel which causes under-fitting of the data, while a large number of frequencies induces a narrow kernel causing over-fitting. With this in mind, we propose a method to smoothly anneal the bandwidth of the NTK by introducing a parameter α that windows the frequency bands of the positional encoding, akin to how coarse-to-fine optimization schemes solve for coarse solutions that are subsequently refined at higher resolutions. We define the weight for each frequency band j as:

$$w_j(\alpha) = \frac{(1 - \cos(\pi \operatorname{clamp}(\alpha - j, 0, 1)))}{2}, \tag{8}$$

where linearly annealing the parameter $\alpha \in [0, m]$ can be interpreted as sliding a truncated Hann window (where the left side is clamped to 1 and the



ground truth rendered color render w/o bg

Figure 7: Our method recovers thin hair strands. By adjusting the camera's far plane, we can render the subject against a flat white background.

right side is clamped to 0) across the frequency bands. The positional encoding is then defined as $\gamma_\alpha(\mathbf{x}) = \big(\mathbf{x}, \cdots, w_k(\alpha)\sin(2^k\pi\mathbf{x}), w_k(\alpha)\cos(2^k\pi\mathbf{x}), \cdots\big)$. During training, we set $\alpha(t) = \frac{mt}{N}$ where t is the current training iteration, and N is a hyper-parameter for when α should reach the maximum number of frequencies m. We provide further analysis in the supplementary materials.

4. Nerfies: Casual Free-Viewpoint Selfies

So far we have presented a generic method of reconstructing non-rigidly deforming scenes. We now present a key application of our system – reconstructing high quality models of human subjects from casually captured selfies, which we dub *"nerfies"*. Our system takes as input a sequence of selfie photos or a selfie video in which the user is standing mostly still. Users are instructed to wave the camera around their face, covering viewpoints within a $45°$ cone. We observe that 20 second captures are sufficient. In our method, we assume that the subject stands against a static background to enable a consistent geometric registration of the cameras. We filter blurry frames using the variance of the Laplacian [36], keeping about 600 frames per capture.

Camera Registration: We seek a registration of the cameras with respect to the static background. We use COLMAP [40] to compute pose for each image and camera intrinsics. This step assumes that enough features are present in the background to register the sequence.

Foreground Segmentation: In some cases, SfM will match features on the moving subject, causing significant misalignment in the background. This is problematic in video captures with correlated frames. In those cases, we found it helpful to discard image features on the subject, which can be detected using a foreground segmentation network.

5. Experiments

5.1. Implementation Details

Our NeRF template implementation closely follows the original [32], except we use a Softplus activation $\ln(1 + e^x)$ for the density. We use a deformation network with

example inputs rendered rgb rendered depth

Figure 8: Our method reconstructs full body scenes captured by a second user with high quality details.

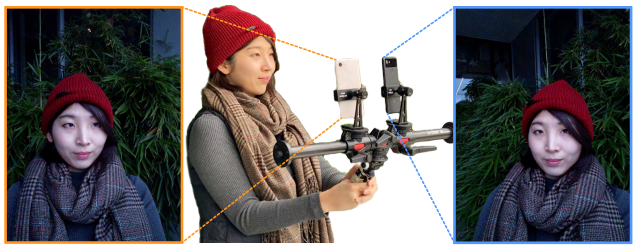


Figure 9: Validation rig used only for evaluation.

depth 6, hidden size 128, and a skip connection at the 4th layer. We use 256 coarse and fine ray samples for full HD (1920×1080) models and half that for the half resolution models. We use 8 dimensions for the latent deformation and appearance codes. For coarse-to-fine optimization we use 6 frequency bands and linearly anneal α from 0 to 6 over 80K iterations. We use the same MSE photometric loss as in NeRF [32] and weight the losses as $L_{\text{total}} = L_{\text{rgb}} + \lambda L_{\text{elastic-r}} + \mu L_{\text{bg}}$ where we use $\lambda = \mu = 10^{-3}$ for all experiments except when mentioned. We train on 8 V100 GPUs for a week for full HD models, and for 16 hours for the half resolution models used for the comparisons in Tab. 1, Fig. 10. We provide more details in the Section A of the appendix.

5.2. Evaluation Dataset

In order to evaluate the quality of our reconstruction, we must be able to measure how faithfully we can recreate the scene from a viewpoint unseen during training. Since we are reconstructing non-rigidly deforming scenes, we cannot simply hold out views from an input capture, as the structure of the scene will be slightly different in every image. We therefore build a simple multi-view data capture rig for the sole purpose of evaluation. We found the multi-view dataset of Yoon *et al.* [52] not representative of many capture scenarios, as it contains too few viewpoints (12) and exaggerated frame-to-frame motions due to temporal subsampling.

Our rig (Fig. 9) is a pole with two Pixel 3's rigidly attached. We have two methods for data capture: (a) for

	Glasses (78 images)		Beanie (74 images)		Curls (57 images)		Kitchen (40 images)		Lamp (55 images)		Toby Sit (308 images)		Mean		Drinking (193 images)		Tail (238 images)		Badminton (356 images)		Broom (197 images)		Mean	
	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓
NeRF [32]	18.1	.474	16.8	.583	14.4	.616	19.1	.434	17.4	.444	22.8	.463	18.1	.502	18.6	.397	23.0	.571	18.8	.392	21.0	.667	20.3	.506
NeRF + latent	19.5	.463	19.5	.535	17.3	.539	20.1	.403	18.9	.386	19.4	.385	19.1	.452	21.9	.233	24.9	.404	20.0	.308	21.9	.576	22.2	.380
Neural Volumes [25]	15.4	.616	15.7	.595	15.2	.588	16.2	.569	13.8	.533	13.7	.473	15.0	.562	16.2	.198	18.5	.559	13.1	.516	16.1	.544	16.0	.454
NSFF†	18.8	.490	18.8	.483	15.5	.569	20.9	.394	17.9	.343	23.3	.391	19.2	.445	23.1	.175	24.2	.363	18.2	.368	22.1	.357	21.9	.316
$\gamma(t)$ + Trans† [23]	22.2	.354	20.8	.471	20.7	.426	22.5	.344	21.9	.283	25.3	.420	22.2	.383	23.7	.151	27.2	.391	22.9	.221	23.4	.627	24.3	.347
Ours ($\lambda = 0.01$)	23.4	.305	22.2	.391	24.6	.319	23.9	.280	23.6	.232	22.9	.159	23.4	.281	22.4	.0872	23.9	.161	22.4	.130	21.5	.245	22.5	.156
Ours ($\lambda = 0.001$)	24.2	.307	23.2	.391	24.9	.312	23.5	.279	23.7	.230	22.8	.174	23.7	.282	21.8	.0962	23.6	.175	22.1	.132	21.0	.270	22.1	.168
No elastic	23.1	.317	24.2	.382	24.1	.322	22.9	.290	23.7	.230	23.0	.257	23.5	.300	22.2	.0863	23.7	.174	22.0	.132	20.9	.287	22.2	.170
No coarse-to-fine	23.8	.312	21.9	.408	24.5	.321	24.0	.277	22.8	.242	22.7	.244	23.3	.301	22.3	.0960	24.3	.257	21.8	.151	21.9	.406	22.6	.228
No SE3	23.5	.314	21.9	.401	24.5	.317	23.7	.282	22.7	.235	22.9	.206	23.2	.293	22.4	.0867	23.5	.191	21.2	.156	20.9	.276	22.0	.177
Ours (base)	24.0	.319	20.9	.456	23.5	.345	22.4	.323	22.1	.254	22.7	.184	22.6	.314	22.6	.127	24.3	.298	21.1	.173	22.1	.503	22.5	.275
No BG Loss	22.3	.317	21.5	.395	20.1	.371	22.5	.290	20.3	.260	22.3	.145	21.5	.296	22.3	.0856	23.5	.210	20.4	.161	20.9	.330	21.8	.196

Table 1: Quantitative evaluation on validation captures against baselines and ablations of our system, we color code each row as **best**, **second best**, and **third best**. †denotes use of temporal information. Please see Sec. 5.3 for more details.

selfies we use the front-facing camera and capture time-synchronized photos using the method of Ansari *et al*. [1], which achieves sub millisecond synchronization; or (b) we use the back-facing camera and record two videos which we manually synchronize based on the audio; we then subsample to 5 fps. We register the images using COLMAP [40] with rigid relative camera pose constraints. Sequences captured with (a) contain fewer frames (40~78) but the focus, exposure, and time are precisely synchronized. Sequences captured with (b) have denser samples in time (between 193 and 356 frames) but the synchronization is less precise and exposure and focus may vary between the cameras. We split each capture into a training set and a validation set. We alternate assigning the left view to the training set, and right to the validation, and vice versa. This avoids having regions of the scene that one camera has not seen.

Quasi-static scenes: We capture 5 human subjects using method (a), that attempt to stay as still as possible during capture, and a mostly still dog using method (b).

Dynamic scenes: We capture 4 dynamic scenes containing deliberate motions of a human subject, a dog wagging its tail, and two moving objects using method (b).

5.3. Evaluation

Here we provide quantitative and qualitative evaluations of our model. However, to best appreciate the quality of the reconstructed *nerfies*, we encourage the reader to watch the supplementary video that contains many example results.

Quantitative Evaluation: We compare against NeRF and a NeRF + latent baseline, where NeRF is conditioned on a per-image learned latent code [5] to modulate density and color. We also compare with a variant of our system similar to the concurrent work of D-NeRF [37], which conditions a translational deformation field with a position encoded time variable $\gamma(t)$ instead of a latent code ($\gamma(t)$+trans in Tab. 1). We also compare with the high quality model of Neural Volumes (NV) [25] using a single view as input to the encoder, and Neural Scene Flow Fields (NSFF) [23]. We do not evaluate the method of Yoon et al. [52] due to the lack of available code (note that NSFF outperforms it). NSFF and the $\gamma(t) + trans$ baseline use temporal information while other baselines and our method do not. NSFF also uses auxilliary supervision such as estimated flow and relative depth maps; we do not. Photometric differences between the two rig cameras may exist due to different exposure/white balance settings and camera response curves. We therefore swap the per-frame appearance code ψ_i for a per-camera $\{\psi_L, \psi_R\} \in \mathbb{R}^2$ instead for validation rig captures.

Tab. 1 reports LPIPS [54] and PSNR metrics for the unseen validation views. PSNR favors blurry images and is therefore not an ideal metric for dynamic scene reconstruction; we find that LPIPS is more representative of visual quality. See Fig. 10 for side-by-side images with associated PSNR/LPIPS metrics. Our method struggles with PSNR due to slight misalignments resulting from factors such as gauge ambiguity [30] while we outperform all baselines in terms of LPIPS for all sequences.

Ablation Study: We evaluate each of our contributions: SE(3) deformations, elastic regularization, background regularization, and coarse-to-fine optimization. We ablate them one at a time, and all at once (Ours (bare) in Tab. 1). As expected, a stronger elastic regularization ($\lambda = 0.01$) improves results for dynamic scenes compared to the baseline ($\lambda = 0.001$) while minimally impacting quasi-static scenes. Removing the elastic loss hurts performance for quasi-static scenes while having minimal effect on the dynamic scene; this may be due to the larger influence of other losses in the presence of larger motion. Elastic regularization fixes distortion artifacts when the scene is under-constrained (e.g., Fig. 4). Disabling coarse-to-fine regularization mildly drops performance for quasi-static scenes while causing a significant drop for dynamic scenes. This is expected since large motions are a main source of local minima (e.g., Fig. 5). Our SE(3) deformations also quantitatively outperform translational deformations. Background regularization helps PSNR by reducing shifts in static regions and removing it performs worse. Finally, removing all of our contributions performs the worst in terms of LPIPS.

Qualitative Results: We show results for the captures used in the quantitative evaluation in Fig. 10. Our method can reconstruct fine details such as strands of hair (e.g., in Curls of Tab. 1 and Fig. 7), shirt wrinkles, and glasses (Fig. 8). Our method works on general scenes beyond human subjects as shown in Fig. 10. In addition, we can create smooth

Figure 10: Comparisons of baselines and our method. PSNR / LPIPS metrics on bottom right with best colored red.

animations by interpolating the deformation latent codes of any input state as shown in Fig. 6.

Elastic Regularization: Fig. 4 shows an example where the user only captured 20 images mostly from one side of their face, while their head tracked the camera. This results in ambiguous geometry. Elastic regularization helps in such under-constrained cases, reducing distortion significantly.

Depth Visualizations: We visualize the quality of our reconstruction using depth renders of the density field. Unlike NeRF[32] that visualizes the expected ray termination distance, we use the *median* depth termination distance, which we found to be less biased by residual density in free space (see Fig. 7). We define it as the depth of the first sample with accumulated transmittance $T_i \geq 0.5$ (Eqn. 3 of NeRF [32]).

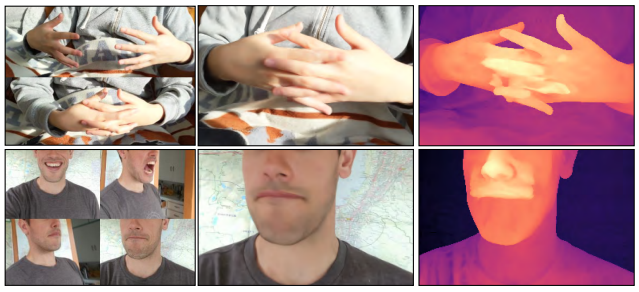


Figure 11: Our method fails in the presence of topological changes. Although the rendered color views look good from some viewpoints, the recovered geometry is incorrect.

Limitations: Our method struggles with topological changes e.g., opening/closing of the mouth (see Fig. 11) and may fail for certain frames in the presence of rapid motion (see supplementary). As mentioned in §3.4, our deformations are unconstrained so static regions may shift; this contributes to the disjunction between PSNR and LPIPS in Tab. 1. Future work may address this by modeling static regions separately as in [23, 29]. Finally, the quality of our method depends on camera registration, and when SfM fails so do we.

6. Conclusion

Deformable Neural Radiance Fields extend NeRF by modeling non-rigidly deforming scenes. We show that our as-rigid-as-possible deformation prior, and coarse-to-fine deformation regularization are the key to obtaining high-quality results. We showcase the application of casual selfie captures (*nerfies*), and enable high-fidelity reconstructions of human subjects using a cellphone capture. Future work may tackle larger/faster motion, topological variations, and enhance the speed of training/inference.

Acknowledgments

We thank Peter Hedman and Daniel Duckworth for providing feedback in early drafts, and all our capture subjects for their patience, including Toby who was a good boy.

References

[1] Sameer Ansari, Neal Wadhwa, Rahul Garg, and Jiawen Chen. Wireless software synchronization of multiple distributed cameras. *ICCP*, 2019. 7

[2] Jonathan T. Barron. A general and adaptive robust loss function. *CVPR*, 2019. 5

[3] Sai Bi, Zexiang Xu, Pratul Srinivasan, Ben Mildenhall, Kalyan Sunkavalli, Miloš Hašan, Yannick Hold-Geoffroy, David Kriegman, and Ravi Ramamoorthi. Neural reflectance fields for appearance acquisition. *arXiv preprint arXiv:2008.03824*, 2020. 2

[4] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3D faces. *SIGGRAPH*, 1999. 2

[5] Piotr Bojanowski, Armand Joulin, David Lopez-Pas, and Arthur Szlam. Optimizing the latent space of generative networks. *ICML*, 2018. 2, 4, 7

[6] James Booth, Anastasios Roussos, Allan Ponniah, David Dunaway, and Stefanos Zafeiriou. Large scale 3D morphable models. *IJCV*, 2018. 2

[7] Sofien Bouaziz, Sebastian Martin, Tiantian Liu, Ladislav Kavan, and Mark Pauly. Projective dynamics: Fusing constraint projections for fast simulation. *ACM TOG*, 2014. 2, 4

[8] Sofien Bouaziz, Yangang Wang, and Mark Pauly. Online modeling for realtime facial animation. *ACM TOG*, 2013. 2

[9] Aljaž Božič, Pablo Palafox, Michael Zollhöfer, Angela Dai, Justus Thies, and Matthias Nießner. Neural non-rigid tracking. *arXiv preprint arXiv:2006.13240*, 2020. 2

[10] Christoph Bregler, Aaron Hertzmann, and Henning Biermann. Recovering non-rigid 3d shape from image streams. *CVPR*, 2000. 2

[11] Thomas J Cashman and Andrew W Fitzgibbon. What shape are dolphins? building 3D morphable models from 2D images. *TPAMI*, 2012. 2

[12] Isaac Chao, Ulrich Pinkall, Patrick Sanan, and Peter Schröder. A simple geometric model for elastic deformations. *ACM Trans. Graph.*, 2010. 2, 4

[13] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *CVPR*, 2019. 2

[14] Alvaro Collet, Ming Chuang, Pat Sweeney, Don Gillett, Dennis Evseev, David Calabrese, Hugues Hoppe, Adam Kirk, and Steve Sullivan. High-quality streamable free-viewpoint video. *ACM ToG*, 2015. 1, 2

[15] Mingsong Dou, Sameh Khamis, Yury Degtyarev, Philip Davidson, Sean Ryan Fanello, Adarsh Kowdle, Sergio Orts Escolano, Christoph Rhemann, David Kim, Jonathan Taylor, et al. Fusion4D: Real-time performance capture of challenging scenes. *ACM ToG*, 2016. 1, 2, 4

[16] Ohad Fried, Ayush Tewari, Michael Zollhöfer, Adam Finkelstein, Eli Shechtman, Dan B Goldman, Kyle Genova, Zeyu Jin, Christian Theobalt, and Maneesh Agrawala. Text-based editing of talking-head video. *ACM TOG*, 2019. 2

[17] Stuart Geman and Donald E. McClure. Bayesian image analysis: An application to single photon emission tomography. *Proceedings of the American Statistical Association*, 1985. 5

[18] Kaiwen Guo, Peter Lincoln, Philip Davidson, Jay Busch, Xueming Yu, Matt Whalen, Geoff Harvey, Sergio Orts-Escolano, Rohit Pandey, Jason Dourgarian, Danhang Tang, Anastasia Tkach, Adarsh Kowdle, Emily Cooper, Mingsong Dou, Sean Fanello, Graham Fyffe, Christoph Rhemann, Jonathan Taylor, Paul Debevec, and Shahram Izadi. The relightables: Volumetric performance capture of humans with realistic relighting. *ACM ToG*, 2019. 1

[19] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *NeurIPS*, 2018. 5

[20] Chiyu Jiang, Jingwei Huang, Andrea Tagliasacchi, Leonidas Guibas, et al. Shapeflow: Learnable deformations among 3d shapes. *arXiv preprint arXiv:2006.07982*, 2020. 2

[21] Hyeongwoo Kim, Pablo Garrido, Ayush Tewari, Weipeng Xu, Justus Thies, Matthias Nießner, Patrick Pérez, Christian Richardt, Michael Zollöfer, and Christian Theobalt. Deep video portraits. *ACM ToG*, 2018. 2

[22] Hao Li, Linjie Luo, Daniel Vlasic, Pieter Peers, Jovan Popović, Mark Pauly, and Szymon Rusinkiewicz. Temporally coherent completion of dynamic shapes. *ACM TOG*, 2012. 2

[23] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. *arXiv preprint arXiv:2011.13084*, 2020. 2, 7, 8

[24] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *NeurIPS*, 2020. 2

[25] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: learning dynamic renderable volumes from images. *ACM ToG*, 2019. 2, 3, 7, 8

[26] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graph.*, 2015. 2

[27] BD LUCAS. An iterative image registration technique with an application to stereo vision. In *Proceedings of Imaging Understanding Workshop, 1981*, 1981. 5

[28] Kevin M Lynch and Frank C Park. *Modern Robotics*. Cambridge University Press, 2017. 4

[29] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. *CVPR*, 2021. 2, 3, 8

[30] Philip F McLauchlan. Gauge invariance in projective 3d reconstruction. In *Proceedings IEEE Workshop on Multi-View Modeling and Analysis of Visual Scenes (MVIEW'99)*, pages 37–44. IEEE, 1999. 7

[31] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3D reconstruction in function space. In *CVPR*, 2019. 2

[32] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. *ECCV*, 2020. 1, 2, 3, 5, 6, 7, 8

[33] Richard A Newcombe, Dieter Fox, and Steven M Seitz. DynamicFusion: Reconstruction and tracking of non-rigid scenes in real-time. *CVPR*, 2015. 2, 3, 4

[34] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Occupancy flow: 4d reconstruction by learning particle dynamics. *ICCV*, 2019. 2

[35] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *CVPR*, 2019. 2

[36] José Luis Pech-Pacheco, Gabriel Cristóbal, Jesús Chamorro-

Martinez, and Joaquín Fernández-Valdivia. Diatom autofocusing in brightfield microscopy: a comparative study. In *ICPR*, volume 3, pages 314–317. IEEE, 2000. 6

[37] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. *arXiv preprint arXiv:2011.13961*, 2020. 2, 7

[38] Olinde Rodrigues. De l'attraction des sphéroïdes. In *Correspondence Sur l'École Impériale Polytechnique*, pages 361–385, 1816. 4

[39] Tanner Schmidt, Richard Newcombe, and Dieter Fox. Dart: dense articulated real-time tracking with consumer depth cameras. *Autonomous Robots*, 2015. 2

[40] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. *CVPR*, 2016. 6, 7

[41] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. GRAF: Generative radiance fields for 3D-aware image synthesis. *NeurIPS*, 2020. 2

[42] Eftychios Sifakis and Jernej Barbic. Fem simulation of 3D deformable solids: A practitioner's guide to theory, discretization and model reduction. *ACM SIGGRAPH 2012 Courses*, 2012. 4

[43] Vincent Sitzmann, Julien N.P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *NeurIPS*, 2020. 2

[44] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3D-structure-aware neural scene representations. In *NeurIPS*, 2019. 2

[45] Olga Sorkine and Marc Alexa. As-rigid-as-possible surface modeling. *EUROGRAPHICS*, 2007. 2, 4

[46] Robert W. Sumner, Johannes Schmid, and Mark Pauly. Embedded deformation for shape manipulation. *ACM TOG*, 2007. 2, 4

[47] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *NeurIPS*, 2020. 2, 3, 5

[48] Lorenzo Torresani, Aaron Hertzmann, and Chris Bregler. Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors. *TPAMI*, 2008. 2

[49] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video, 2021. 2

[50] Wenqi Xian, Jia-Bin Huang, Johannes Kopf, and Changil Kim. Space-time neural irradiance fields for free-viewpoint video. *arXiv preprint arXiv:2011.12950*, 2020. 2

[51] Hongyi Xu, Eduard Gabriel Bazavan, Andrei Zanfir, William T Freeman, Rahul Sukthankar, and Cristian Sminchisescu. GHUM & GHUML: Generative 3D human shape and articulated pose models. *CVPR*, 2020. 2

[52] Jae Shin Yoon, Kihwan Kim, Orazio Gallo, Hyun Soo Park, and Jan Kautz. Novel view synthesis of dynamic scenes with globally coherent depths from a monocular camera. In *CVPR*, 2020. 2, 6, 7

[53] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020. 2

[54] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 7

[55] Michael Zollhöfer, Matthias Nießner, Shahram Izadi, Christoph Rehmann, Christopher Zach, Matthew Fisher, Chenglei Wu, Andrew Fitzgibbon, Charles Loop, Christian Theobalt, et al. Real-time non-rigid reconstruction using an RGB-D camera. *ACM Trans. Graph.*, 2014. 4

[56] Michael Zollhöfer, Justus Thies, Pablo Garrido, Derek Bradley, Thabo Beeler, Patrick Pérez, Marc Stamminger, Matthias Nießner, and Christian Theobalt. State of the art on monocular 3D face reconstruction, tracking, and applications. *Computer Graphics Forum*, 2018. 2

[57] Silvia Zuffi, Angjoo Kanazawa, David W Jacobs, and Michael J Black. 3D menagerie: Modeling the 3D shape and pose of animals. *CVPR*, 2017. 2