Nerfies: Deformable Neural Radiance Fields

Keunhong Park1* Utkarsh Sinha2 Jonathan T. Barron2 Sofien Bouaziz2
Dan B Goldman2 Steven M. Seitz1,2 Ricardo Martin-Brualla2

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

![Image](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, viewpoint-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
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]. Dynamic Fusion [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 been 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.
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 \( F : (x, d, \psi, t) \rightarrow (c, \sigma) \) which maps a 3D position \( x = (x, y, z) \) and viewing direction \( d = (\phi, \theta) \) to a color \( c = (r, g, b) \) and density \( \sigma \). In practice, NeRF maps the inputs \( x \) and \( d \) using a sinusoidal positional encoding \( \gamma : \mathbb{R}^3 \rightarrow \mathbb{R}^{3+4m} \) defined as

\[
\gamma(x) = (x, \cdots, \sin(2^k \pi x), \cos(2^k \pi x), \cdots),
\]

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 \( x \in \mathbb{R}^3 \) 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 \( \psi \) 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 \( T_i : x \rightarrow x' \) that maps all observation-space coordinates \( x \) to a canonical-space coordinate \( x' \). We model the deformation fields for all time steps using a mapping \( T : (x, \omega_i) \rightarrow x', \) which is conditioned on a per-frame learned latent deformation code \( \omega_i \). 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:

\[
G(x, d, \psi_i, \omega_i) = F(T(x, \omega_i), d, \psi_i).
\]

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 \( V : (x, \omega_i) \rightarrow t \), defining the transformation as \( T(x, \omega_i) = x + V(x, \omega_i) \). 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.

![Figure 2: We associate a latent deformation code (ω) and an appearance code (ψ) 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 ω. We query the template NeRF [32] using the transformed sample (x′, y′, z′), viewing direction (θ, φ) and appearance code ψ as inputs to the MLP and integrate samples along the ray following NeRF.](image)
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 underconstrained 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 \(\omega_i\), 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 \(J_T(x)\) of this mapping at a point \(x \in \mathbb{R}^3\) describes the best linear approximation of the transformation at that point. We can therefore control the local behavior of the deformation through \(J_T\) \[42\]. Note that unlike other approaches using discretized surfaces, our continuous formulation allows us to directly compute \(J_T\) through automatic differentiation of the MLP. There are several ways to penalize the deviation of the Jacobian \(J_T\) from a rigid transformation. Considering the singular-value decomposition of the Jacobian \(J_T = U \Sigma V^T\), multiple approaches \[7, 12\] penalize the deviation from the closest rotation as \(\|J_T - R\|_F\), where \(R = VU^T\) and \(\|\cdot\|_F\) is the Frobenius norm. We opt to directly work with the singular values of \(J_T\) 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:

\[
L_{\text{elastic}}(x) = \|\log \Sigma - \log I\|_F^2 = \|\log \Sigma\|_F^2 ,
\]

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

\[
L_{\text{elastic}}(x) = \|\log \Sigma - \log I\|_F^2 = \|\log \Sigma\|_F^2 ,
\]

where log here is the matrix logarithm.
defined above using a robust loss:

\[
L_{\text{elastic}}(x) = \rho \left( ||\log \Sigma||_F, c \right), \quad (5)
\]

\[
\rho(x, c) = \frac{2(x/c)^2}{{(x/c)}^2 + 4}, \quad (6)
\]

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 \(\{x_1, \ldots, x_K\}\), we penalize movement as:

\[
L_{bg} = \frac{1}{K} \sum_{k=1}^{K} ||T(x_k) - x_k||_2. \quad (7)
\]

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 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 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 \(\alpha\) 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:

\[
w_j(\alpha) = \frac{1 - \cos(\pi \text{clamp}(\alpha - j, 0, 1))}{2}, \quad (8)
\]

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
right side is clamped to 0) across the frequency bands. The positional encoding is then defined as \( \gamma_\alpha(x) = (x, \cdots, w_k(\alpha) \sin(2^k \pi x), w_k(\alpha) \cos(2^k \pi x), \cdots) \). During training, we set \( \alpha(t) = \frac{t}{N} \) where \( t \) is the current training iteration, and \( N \) is a hyper-parameter for when \( \alpha \) 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 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 \( \alpha \) 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
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] manually synchronize based on the audio; we then subsample to 5 fps. We register the images using COLMAP [40] to slight misalignments resulting from factors such as gauge

### 5.3. Evaluation

| 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. \(^\dagger\) denotes use of temporal information. Please see Sec. 5.3 for more details. |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| GLASSES | BEANIE | CURLS | KITCHEN | LAMP | TORY SIT | DRINKING | TAIL | BADMINTON | BROOM |
| Glasses (78 images) | Beanie (74 images) | Curls (57 images) | Kitchen (40 images) | Lamp (55 images) | Tory Sit (108 images) | Drinking (193 images) | Tail (238 images) | Badminton (156 images) | Broom (197 images) |
| NeRF [15] | 18.1 | 474 | 16.8 | 584 | 14.4 | 616 | 19.1 | 434 | 17.4 | 444 | 22.8 | 463 | 18.1 | 502 | 18.6 | 493 | 21.0 | 774 | 18.8 | 492 | 20.1 | 667 | 20.3 | 506 |
| NeRF + latent | 19.5 | 463 | 19.5 | 535 | 17.3 | 539 | 20.1 | 403 | 18.9 | 396 | 19.4 | 395 | 19.1 | 452 | 21.9 | 233 | 23.3 | 404 | 20.9 | 308 | 21.9 | 576 | 22.2 | 300 |
| NeRF + latent, Real Neural Volumes (25) | 15.4 | 616 | 15.7 | 595 | 15.2 | 598 | 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 | 481 | 15.5 | 569 | 20.9 | 394 | 17.9 | 342 | 23.3 | 391 | 19.2 | 445 | 23.1 | 175 | 24.2 | 363 | 19.2 | 368 | 22.1 | 357 | 21.9 | 316 |
| (γ) + Trans [23] | 22.2 | 354 | 20.8 | 471 | 20.7 | 428 | 22.3 | 344 | 21.9 | 283 | 25.3 | 420 | 22.2 | 335 | 23.1 | 115 | 22.9 | 303 | 22.4 | 221 | 21.5 | 327 | 24.2 | 341 |
| Ours (λ = 0.01) | 23.4 | 308 | 22.2 | 391 | 24.6 | 319 | 23.7 | 280 | 23.6 | 232 | 22.9 | 315 | 23.4 | 284 | 22.4 | 087 | 21.9 | 186 | 22.1 | 130 | 21.5 | 146 | 22.5 | 325 |
| Ours (λ = 0.001) | 24.2 | 307 | 23.2 | 394 | 24.9 | 312 | 23.5 | 279 | 23.7 | 238 | 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 | 23.9 | 290 | 23.7 | 238 | 23.0 | 257 | 23.3 | 300 | 22.2 | 0953 | 24.7 | 174 | 22.0 | 162 | 20.9 | 267 | 22.2 | 170 |
| No coarse-to-fine | 23.8 | 318 | 21.9 | 405 | 24.5 | 332 | 24.0 | 377 | 22.7 | 242 | 22.7 | 244 | 23.1 | 301 | 22.3 | 0999 | 24.3 | 275 | 21.8 | 151 | 21.9 | 407 | 22.6 | 228 |
| No SEI | 23.5 | 314 | 21.9 | 401 | 24.5 | 337 | 23.7 | 282 | 22.7 | 235 | 22.9 | 206 | 23.2 | 253 | 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 | 445 | 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 | 385 | 20.1 | 371 | 22.5 | 290 | 20.3 | 266 | 22.3 | 316 | 21.5 | 296 | 22.3 | 0856 | 23.5 | 218 | 20.4 | 161 | 20.9 | 330 | 21.8 | 196 |
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]).

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


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


