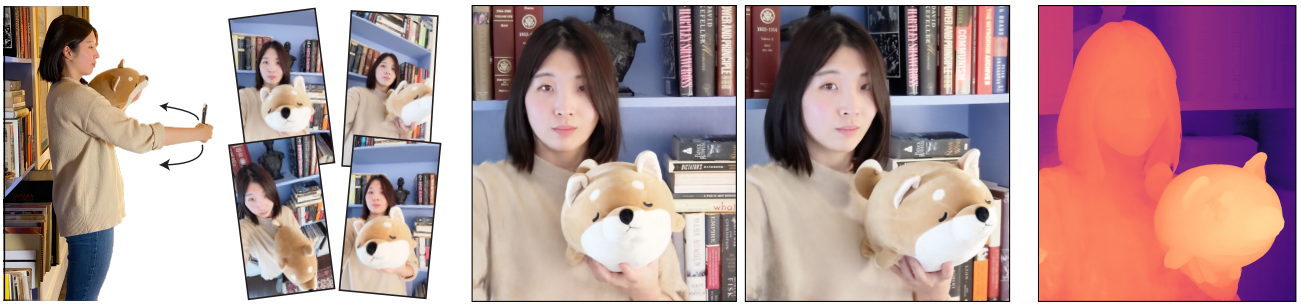
Deformable Neural Radiance Fields

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 renderings (c) and accurate scene geometry (d). Please see supplementary for video results.

Abstract

非刚性运动场景

We present the first method capable of photorealistically reconstructing a non-rigidly deforming scene using photos/videos captured casually from mobile phones. Our approach – D-NeRF– *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,* 弹性正则化 *we propose an elastic regularization of the deformation field that further improves robustness. We show that* D-NeRF *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 data using a rig with two mobile phones that take time-synchronized photos, 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

自拍

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., [16, 17, 21]. 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) [36] 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 [36] 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 [8, 13, 51, 52]. 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 against ground truth, 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 evaluation data at `nerfies.github.io`.

In summary, our contributions are: ① an extension to NeRF to handle non-rigidly deforming objects that optimizes a deformation field per observation; ② volume-preserving 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

Our work intersects the fields of non-rigid reconstruction, model-based reconstruction and neural rendering.

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 [11, 56], making the problem highly ambiguous. Using multi-view captures [16, 17] simplified the problem to one of registering and fusing 3D scans [27]. DynamicFusion [37] 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 [10, 43]. 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 [29] learns a 3D representation of a deformable scene using a voxel grid and warp field regressed from a 3D CNN. Their method uses dozens of synchronized cameras to capture a dynamic scene and we show in our evaluation that it does not extend to dynamic scenes captured from a single camera. OccFlow [39] uses a flow-field to represent 3D human motion over time using an ODE, while our approach does not rely on temporal information. ShapeFlow [23] models shapes as deformations from a set of canonical models, using a divergence-free parameterization of the deformation field. We instead use an elastic regularization of the deformation field.

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 [5, 7, 9], human bodies [30, 58], and animals [12, 64]. Although some methods show impressive results in monocular face reconstruction from color and RGBD cameras [63], such models often lack detail (e.g., hair for humans), or do not model certain aspects of a category (e.g., eyewear or garments for humans). Recently, image translation networks have been applied to improve the realism of composited facial edits [18, 25]. 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. However, our method's lack of semantic understanding limits it from synthesizing unseen states e.g., our method cannot render a smile if it has never seen one.

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 [15, 34, 40] and scenes [36, 50]. Of particular interest are NeRFs [36], that use periodic positional encoding layers [48, 53] to increase resolution, and whose formulation has be extended to handle different lighting conditions [4, 33], transient objects [33], large scenes [28, 60] and to model object categories [46]. Our work focuses on extending NeRFs to handle non-rigid scenes.

Neural Representations: Besides coordinate-based models, neural representations are commonly used for 3D modeling and rendering [54], ranging from voxel grids [24, 29, 38, 41, 49], neural textures [55], and point clouds [1]. A common strategy is to use re-rendering networks to produce more realistic results from proxy renders [18, 25, 32, 35, 59]. However, most of these methods have limited resolution in 3D due to memory constraints, or lack view consistency due to the use of 2D CNNs to render output views.

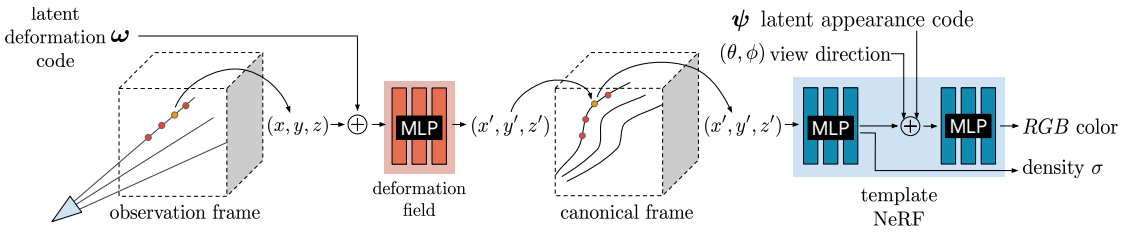


Figure 2: We associate a latent deformation code ($\boldsymbol{\omega}$) and an appearance code ($\boldsymbol{\psi}$) to each image. We trace the camera rays in the observation frame and transform samples along the ray to the canonical frame using a deformation field encoded as an MLP that is conditioned on the deformation code $\boldsymbol{\omega}$. We query the template NeRF module using the transformed sample location (x', y', z'), the viewing direction (θ, ϕ) and the appearance code $\boldsymbol{\psi}$ as inputs to the MLP and integrate samples along the ray following Mildenhall *et al*. [36].

3. Deformable Neural Radiance Fields

Here we describe D-NeRF– 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) [36] (Sec. 3.1) and a per-observation deformation field (Sec. 3.2) that associates a point in observation coordinates to a point on the template. 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 (Sec. 3.3), a background regularization (Sec. 3.4), and a continuous, coarse-to-fine annealing technique that avoids bad local minima (Sec. 3.5).

3.1. Neural Radiance Fields

将3D位置和视角方向映射为颜色和密度

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. Similar to the NeRF-A model [33], we also provide an appearance 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. Coupled with volume rendering techniques, NeRFs can represent scenes with photo-realistic quality. For this reason, we build upon NeRF to tackle the problem of photo-realistic human capture.

两个相机的交线是相同颜色
但是人不是静止的

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 scenes with static structure. Unfortunately, we have found that people do not possess the ability to stay still. This is easy to verify: try to take a selfie video while staying completely still. You will find that our gaze naturally follows the camera, and that even parts we think are still move relative to the background.

3.2. Neural Deformation Fields

通过扩展NeRF可以重构非刚体形变场景
不直接用NeRF，而是作为一个规范模板，渲染时用非刚体变换版本
DF和NV对模板和每帧变换建模，但变换是定义在网格点和体素上的，而我们用MLP建模为连续函数

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 [37] and Neural Volumes [29] 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 $\mathbf{x}'$. In practice, we model the deformation fields for all time steps using a single MLP $T : (\mathbf{x}, \boldsymbol{\omega}_i) \rightarrow \mathbf{x}'$, which is conditioned on a per-frame learned latent 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:

观察到规范的变形

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

渲染时采样观察点并映射到标准模板

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 points on the template, see Fig. 2.

The simplest version of the deformation uses a translational vector field $V : (\mathbf{x}, \boldsymbol{\omega}_i) \rightarrow \mathbf{t}$, defining the deformation 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 chunks of the scene 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 an SE(3) transform as a rotation $\mathbf{q}$ with pivot point $\mathbf{s}$ followed by a translation $\mathbf{t}$. The pivot allows the network

旋转向量+位移向量

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.

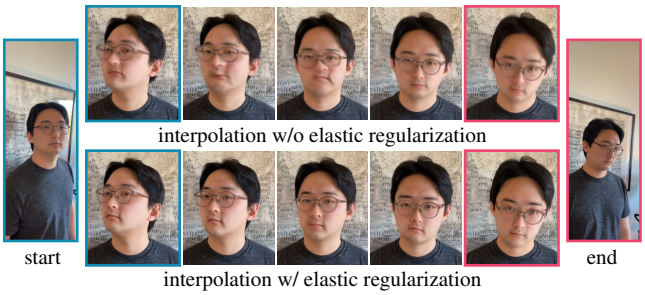


Figure 4: Novel views synthesized without and with elastic regularization by linearly interpolating observation deformation codes (keyframes are outlined). Without elastic regularization, intermediate states show distortions.

to rotate groups of points distant from the origin without relying on the translation. We encode the rotation as a pure log-quaternion $\mathbf{p} = (0, \mathbf{v})$ whose exponential is guaranteed to be a unit quaternion and hence a valid rotation:

$$\mathbf{q} = \exp(\mathbf{p}) = \begin{pmatrix} \cos\|\mathbf{v}\| \\ \frac{\mathbf{v}}{\|\mathbf{v}\|}\sin\|\mathbf{v}\| \end{pmatrix} . \tag{2}$$

Note that this can also be seen as an axis-angle representation where $\mathbf{v}/\|\mathbf{v}\|$ is the unit axis of rotation and $2\|\mathbf{v}\|$ is the angle of rotation. The deformation using the SE(3) transformation is then given by:

$$\mathbf{x}' = \mathbf{q}(\mathbf{x} - \mathbf{s})\mathbf{q}^{-1} + \mathbf{s} + \mathbf{t} . \tag{3}$$

As mentioned before, we encode the transformation field in an MLP $W : (\mathbf{x}, \boldsymbol{\omega}_i) \rightarrow (\mathbf{v}, \mathbf{s}, \mathbf{t})$ 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 as proposed by Bojanowski *et al*. [6]. An important property of the log-quaternion is that $\exp(\mathbf{0})$ is the identity. 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 underconstrained optimization problems which yield implausible results and artifacts (see Fig. 4). It is therefore crucial to introduce priors that lead to a more plausible solution.

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 [8, 13, 51, 52]. In the vision community, these energies have been extensively used for the reconstruction and tracking of non-rigid scenes and objects [17, 37, 62] 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 continuous 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 non-linear 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 our deformation through the Jacobian of T [47]. Note that unlike other approaches using discretized surfaces, our continuous formulation allows us to directly compute the Jacobian of this mapping 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 [8, 13] 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:

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

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 defined above using a robust loss:

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

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

纯对数-四元数，指数是单位四元数

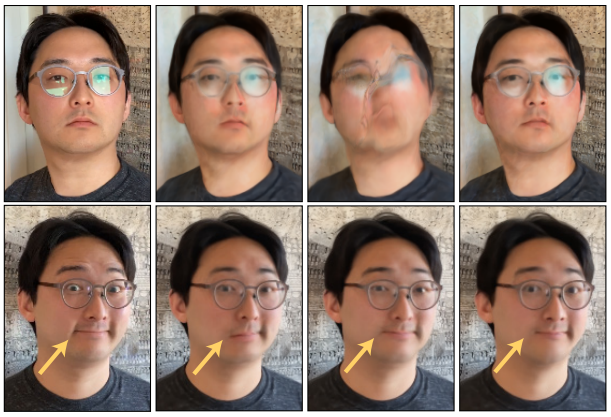
也常用于离散表面，如mesh

雅克比矩阵逼近非线性映射

向后移动等价于size缩小有很多解这会引入优化问题和伪像

几何处理和物理仿真，常用elastic 能量函数衡量局部形变与刚性运动的差别，常用语重建和跟踪非刚体场景和目标

人体移动如人脸表情变化会破坏局部刚性假设，因此用鲁棒损失重新映射上述弹性能量函数



ground truth | 4 frequencies | 8 frequencies | coarse-to-fine

Figure 5: Effect of number of frequencies used to model deformation: With 4 frequencies the head is correctly rotated (top) but the smile is not captured (bottom). With 8 frequencies, the smile is captured but the face is blurry when rotated. Our coarse-to-fine approach works well in both cases.

4频率头部转动正确，笑脸不对，8频率笑脸对，转动模糊

where $\rho(\cdot)$ is the Geman-McClure robust error function [19] implemented as per Barron [3] with hyperparameter $c = 0.03$. The robust error function causes the gradients of the loss to fall off to zero for large values of the argument, reducing the influence of outliers during training.

鲁棒损失对非常大的值梯度为0，减少离群点的影响

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 [36].

每个仿射点都有一个正则项，因为目标在某些位置非刚体运动

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:

背景是静止的，因此加正则项防止形变，同时对齐模板帧与观察帧

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

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 core component of the NeRF architecture is the positional encoding. We employ a similar concept for our deformation field MLP and use a function $\gamma : \mathbb{R}^3 \rightarrow \mathbb{R}^{3+6m}$ defined as $\gamma(\mathbf{x}) = \left(\mathbf{x}, \cdots, \sin\left(2^k\pi\mathbf{x}\right), \cos\left(2^k\pi\mathbf{x}\right), \cdots\right)$ with $k \in \{0, \ldots, m-1\}$. This function projects a positional vector $\mathbf{x} \in \mathbb{R}^3$ to a high dimensional space using a set of sine and cosine functions of increasing frequencies. The hyper-parameter m controls the number of frequency bands (and therefore the highest frequency) used in the mapping. This has been shown to control the smoothness of the network [53]: A higher value of m allows higher frequency details to be modeled, but also may result in NeRF overfitting and modeling image noise as 3D structure.

采用MLP投影到高维三角函数空间以提升频率

超参数m控制网络的光滑性，m越大允许的频率越高，也更容易更拟合

We observe that jointly optimizing a NeRF together with a deformation field leads to an optimization problem that is prone to local minima. Early in training, neither the NeRF nor the deformation field contain meaningful information. If we use a large value for m, this means that our deformation field can over-fit to an incomplete NeRF template. For example, if a subject rotates their head sideways, a network using a large m would often choose to keep the head in the forward position and encode changes of appearance using the view direction component of NeRF. On the other hand, if we use a small value for m, the network will be unable to model deformations which require high frequency details such as facial expressions or moving strands of hair.

刚开始联合训练两者会造成过拟合

m太大导致都是正脸，只有外观变化

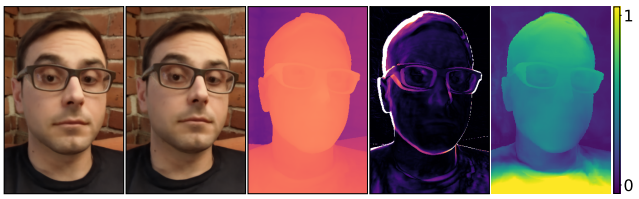
m太小无法建模高频信息

Recently, Tancik *et al.* [53] showed that the positional encoding used in NeRF has a convenient interpretation in terms of the neural tangent kernel (NTK) [22] of NeRF's MLP: a stationary interpolating kernel where m controls a tunable "bandwidth" of that interpolating 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:

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

m越小，核带宽越大，导致欠拟合，越大带宽越窄，过拟合

引入alpha平滑退火NTK的带宽，类似于coarse2fine的优化方式

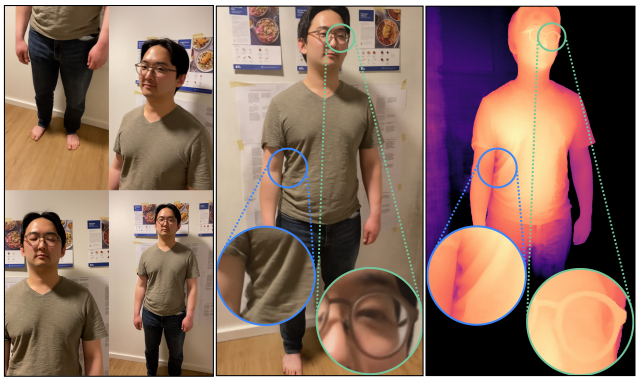


template color | observation color | observation depth | depth difference | deformation mag (cm)

Figure 6: Users move even when trying not to. We show the template and an observed state from the same viewpoint, the difference in depth, and the deformation magnitude of the observed state (more than 0.5 cm for most of the face).

ground truth rendered color render w/o background

Figure 7: *Nerfies* can reconstruct thin hair strands. We can render a subject without the background by adjusting the far plane of the camera. The background is not perfectly white due to residual density along the volume.



example inputs rendered rgb rendered depth

Figure 8: Although not technically *nerfies*, our method reconstructs full body scenes captured by a second user with high quality details.

input images ground truth rendered color rendered depth

Figure 9: Not relying on domain specific priors enables our method to reconstruct any deformable object. In this case, our dog Toby fails to stay still, yet we recover an accurate model.

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 $\gamma_\alpha(\mathbf{x}) = \big(\mathbf{x}, \cdots, w_k(\alpha)\sin\left(2^k\pi\mathbf{x}\right), w_k(\alpha)\cos\left(2^k\pi\mathbf{x}\right), \cdots\big)$. 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.

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 for still subjects, while minute long captures can reconstruct subjects who are deliberately moving. In our method, we assume that the subject stands against a static background to enable a consistent geometric registration of the cameras.

Frame Selection: We filter blurry input frames using the variance of the Laplacian [42] and select around 600 frames per capture.

Camera Registration: We seek a registration of the cameras with respect to the static background. We use structure-from-motion (SfM) [44, 45] to compute camera poses for each image and intrinsic calibration, including distortion coefficients. This step assumes that sufficient features are present in the background to register the sequence. Wider field-of-view (FoV) cameras capture more features and thus using the front-facing camera is preferred due to its typically larger FoV. For convenience, we scale the scene to approximate metric units using estimated facial landmarks from MediaPipe's face mesh [31] and assuming an average interpupillary distance of 6cm.

根据瞳距6cm缩放场景到公制单位

用SfM计算相机位置和内参，以及畸变系数

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 were detected using a foreground segmentation network.

5. Experiments

5.1. Implementation Details

Our NeRF template implementation closely follows the original [36], 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 code. For coarse-to-fine optimization we use 8 frequency bands and linearly anneal α from 0 to 8 over 80K iterations. We use the same MSE photometric loss as in NeRF [36] 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. We train on 8 V100 GPUs for a week for full HD models, and

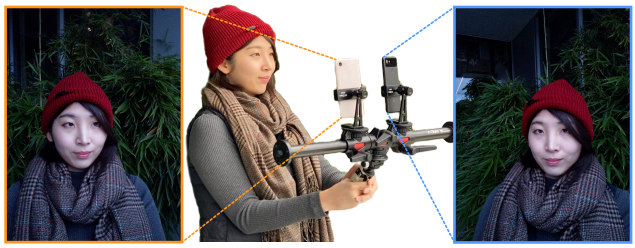


Figure 10: Our *validation* rig consists of two Pixel 3 phones fixed on a rail 15cm apart. It is only used for evaluation.

for 16 hours for the half resolution models used for the comparisons in Tab. 1, and Fig. 12. We provide more details in the Section A of the appendix and plan to release code and data at nerfies.github.io.

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 emphasize that this rig is not necessary for our method to function, and is only used to obtain ground truth to compare our reconstruction against.

Our rig is a simple pole with two Pixel 3 phones attached using a selfie tripod mount. We use the selfie cameras of the phones and capture time-synchronized photos using the method of Ansari *et al*. [2], which achieves sub millisecond synchronization. We register the images using COLMAP [44] with rigid relative camera pose constraints.

We capture 5 subjects and split each capture into a training set and a validation set. Each sequence contains between 40 and 78 frames. We alternate assigning the left view to the training set, and right to the validation, and vice versa. This is often necessary to avoid having regions of the scene that one camera has not seen. We only capture still photos from the validation rig so there is no concern of temporal dependence between images.

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 available at nerfies.github.io, that contains renders of our reconstructions along smooth camera trajectories.

Quantitative Evaluation: We compare against NeRF and a NeRF + latent baseline, where NeRF is conditioned on a per-image learned latent code [6] to modulate density and color. We also compare with the high quality model of Neural Volumes [29] using a single view as input to the encoder. Our validation rig captures images with locked exposure and white balance, but small photometric differences between the two rig cameras still exist. We therefore change the per-frame appearance code ψ_i to be per-camera $\{\psi_L, \psi_R\} \in \mathbb{R}^2$ instead. Table 1 reports LPIPS [61], MS-SSIM [57], and PSNR metrics for the unseen validation views. Our model improves upon the baselines in all metrics and sequences.

example inputs novel views for same deformation code

Figure 11: If the user's gaze consistently follows the camera, the reconstructed *nerfie* represents the user's gaze as geometry, akin to the Hollow-Face illusion [20]. This is apparent in the depth map and makes the reconstructed model appear as if they are looking at the camera even when the geometry is fixed.

Ablation Study: We also evaluate the individual contributions of the features of D-NeRF: elastic regularization, background regularization, and coarse-to-fine optimization. We ablate them one at a time, and all of them simultaneously – a model we refer to as 'Ours (base)'. Background regularization helps the most by reducing shifting of the background, which is emphasized by the full image comparisons. Coarse-to-fine optimization leads to sharper reconstruction, while adding elastic regularization leads to fewer deformation artifacts, whose effects are not well captured by metrics.

Qualitative Results: We show results for the captures used in the quantitative evaluation in Fig. 12. D-NeRF can reconstruct fine details such as strands of hair (e.g., in Curly Hair of Tab. 1 and Fig. 7), shirt wrinkles, and glasses (Fig. 8). Our method can also reconstruct non-humans like Toby the dog (Fig. 9), where we only modify our system to use a segmentation model [14] trained on dogs to aid camera registration.

Elastic Regularization: Our method learns a latent space of the deformations of the scene. In Fig. 4, we show interpolations in that space, where elastic regularization is needed to recover a natural head rotation between poses.

Coarse-to-fine Deformation Regularization: In Fig. 5, we compare our method with baselines using a fixed number of frequency bands, that show both under- and over-fitting for too few and for many frequency bands respectively.

Depth Visualizations: We visualize the quality of our reconstruction using depth renders of the density field. Unlike NeRF[36] 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

	GLASSES (78 images)			BEANIE (74 images)			CURLS (57 images)			KITCHEN (40 images)			LAMP (55 images)			MEAN		
	PSNR↑	MS-SSIM↑	LPIPS↓	PSNR↑	MS-SSIM↑	LPIPS↓	PSNR↑	MS-SSIM↑	LPIPS↓	PSNR↑	MS-SSIM↑	LPIPS↓	PSNR↑	MS-SSIM↑	LPIPS↓	PSNR↑	MS-SSIM↑	LPIPS↓
NeRF [36]	17.69	.5962	.4723	16.58	.5524	.5884	14.28	.4517	.5921	18.79	.6873	.4094	17.42	.6447	.4268	16.95	.5865	.4978
NeRF + latent	21.76	.8201	.3239	20.89	.7711	.4235	22.20	.8040	.3446	21.24	.8212	.3075	20.63	.8489	.2364	21.34	.8131	.3272
Neural Volumes [29]	15.62	.5217	.5759	15.82	.5807	.5630	15.26	.5421	.5506	14.84	.5533	.5719	13.56	.5194	.5558	15.02	.5434	.5635
Ours	24.78	.8783	.2354	23.04	.8338	.3444	24.08	.8613	.2526	23.48	.8759	.2299	22.08	.8729	.1807	23.49	.8644	.2486
No elastic	24.61	.8760	.2357	23.22	.8356	.3451	23.75	.8527	.2547	23.28	.8729	.2393	21.96	.8726	.1801	23.36	.8620	.2510
No coarse-to-fine	23.51	.8434	.2551	21.41	.7875	.3684	23.08	.8284	.2939	23.11	.8667	.2455	22.51	.8751	.1876	22.72	.8402	.2701
No background reg.	24.20	.8656	.2360	19.47	.6989	.3904	20.73	.7620	.2964	21.83	.8395	.2569	19.82	.8078	.2061	21.21	.7947	.2772
Ours (base)	23.91	.8479	.2711	21.83	.7816	.4046	22.85	.8224	.3069	22.21	.8209	.3049	21.92	.8571	.2202	22.54	.8260	.3015

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**. Please see Sec. 5.3 for more details.

(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 [36]).
Limitations: The quality of our method depends on the camera registration, and when SfM fails, it cannot recover. Reconstructions sometimes contain artifacts in areas observed less frequently, like the background in the KITCHEN sequence (see Fig. 12). Our method also fails to model topological change such as the opening and closing of the mouth, which tend to be encoded as view-dependent changes instead of geometry (see the supplementary video for an example). In addition, our method cannot resolve ambiguities arising from correlations in the input data, such as when a user's gaze follows the camera, and our reconstructions encode their moving gaze into the geometry (see Fig. 11).

6. Conclusion

Deformable Neural Radiance Fields extend NeRF by modeling non-rigidly deforming scenes. We show that our elastic deformation priors and coarse-to-fine deformation regularization are the key to obtaining high-quality results. We showcase the application of casual selfie captures, which we dub *nerfies*, and enable high-fidelity reconstructions of human subjects using a cellphone capture. Future work includes handling larger deformations, including full body motions, and extending the framework to static camera captures under variable lighting, where geometry can be recovered from lighting changes.

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.

A. Implementation Details

Architecture Details: We provide architecture details of the deformation field network and canonical NeRF networks in Figures 13 and 14 respectively.
Training: We train our network using the Adam optimizer [26] with a learning rate exponentially decayed by a factor 0.1 until the maximum number of iterations is reached. The exact hyper-parameters for each configuration are provided in Tab. 2.
Elastic Regularization: For experiments using the elastic regularization, after 50k iterations with weight 1e-3, we anneal the weight of the weight of the loss to 1e-8 over 50k iterations using a cosine easing curve.
Background Regularization: Since the total number of background points varies per scene, we sample 16000 points for each iteration when computing the background regularization loss in order to avoid memory issues. We additionally jitter each input point using Gaussian noise $\varepsilon \sim \mathcal{N}(0, 0.001)$.

B. Details of Coarse-to-Fine Optimization

Window Function: Our coarse-to-fine deformation regularization is implemented by windowing the frequency bands of the positional encoding. Eqn. 8 of the main paper defines this windowing function as a weight applied to each frequency band. We visualize our windowing function for different values of α in Fig. 15.
NTK: We also show a visualization of the neural tangent kernel (NTK) induced by our annealed positional encoding in Fig. 16. This figure shows the normalized NTK for an 8 layer MLP of width 256. Note how the bandwidth of the interpolation kernel gets narrower as the value of α increases.

C. Evaluation Details

Comparison to Neural Volumes: Neural Volumes [29] reconstructs a deformable model of a subject captured by dozens of time-synchronized cameras. To apply it to our setting, where only one camera sees the subject at each time instance, we modify the encoder to network to take a single input image instead of three, as in the original method. We disable the background estimation branch and learn instead the complete scene centered around the face and scaled to a unit cube. For each frame, we render the volume from the viewpoint of the second camera of the validation rig and compute image comparison metrics. We provide quantitative comparisons in Table 1 in the main paper, and qualitative comparisons in Fig. 12.

We use a 128^3 voxel grid, a 32^3 warp field and train the

Figure 12: Comparison of novel view synthesis results. NeRF variants use the half-resolution models used in Tab. 1. Only our method produces realistic renderings, which are sharper compared to our base model (details in Sec. 5.3).

network for 100k iterations for each of the five sequences. We evaluate all results using the same camera parameters and spatial resolution. We show some renderings when interpolating the camera position between training and validation views in the supplementary video.

D. Dataset Processing

Blurry Frame Filtering: For video captures, we filter blurry frames using the variance of the Laplacian [42]. To compute the blur score for an image, we apply the Laplace operator with kernel size 3 and compute the variance of the

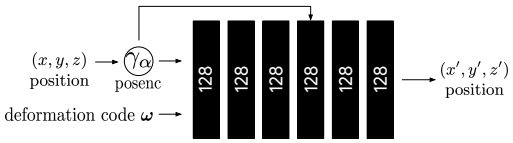


Figure 13: A diagram of our deformation network. The deformation network takes a position encoded position $\gamma_\alpha(\mathbf{x})$ using our coarse-to-fine annealing parameterized by α, along with a deformation code $\boldsymbol{\omega}$ and outputs a deformed position $\mathbf{x}'$. The architecture is identical for all of our experiments.

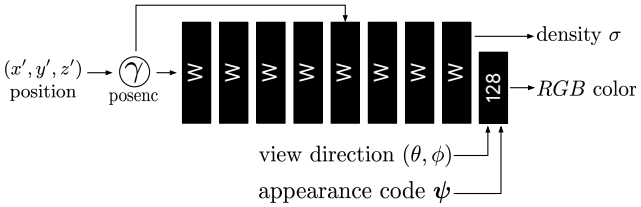


Figure 14: A diagram of the canonical NeRF network. Our network is identical to the original NeRF MLP, except we provide an appearance latent code $\boldsymbol{\psi}$ along with the view direction to allow modulating the appearance as in the NeRF-A model of [33]. The width W of the network is defined according to Tab. 2.

resulting image. We then filter the images based on this score to leave around 600 frames for each capture.

Camera Registration: For camera registration, we first compute a foreground mask using a semantic segmentation network such as DeepLabV3 [14]. We then use COLMAP [44] to compute the camera registration while using the mask to ignore foreground pixels when computing features. We found that this step can improve the quality of the camera registration in the presence of a moving foreground.

Facial Landmarks: Although not necessary for our method, we use facial landmarks for selfie and full body captures to estimate a canonical frame of reference. Using this canonical frame of reference, we automatically generate visually appealing novel view trajectories of our reconstructed *nerfies*, like figure-eight camera paths in front of the user. We compute the 2D facial landmarks using MediaPipe's face mesh [31], and triangulate them in 3D using the Structure-from-Motion camera poses. We then set our canonicalized

Config	Resolution	Steps	Learning Rate	Batch Size	# Samples Fine	# Samples Coarse	Width W
FULL	1080p	1M	7.5e-4	3072	256	256	256
HALF	540p	100K	1e-3	8096	128	128	128

Table 2: Here we provide the hyper-parameters used for each configuration. FULL is the full resolution configuration used in our qualitative results. HALF is half the resolution of FULL and is used for our quantitative evaluation and ablation studies.

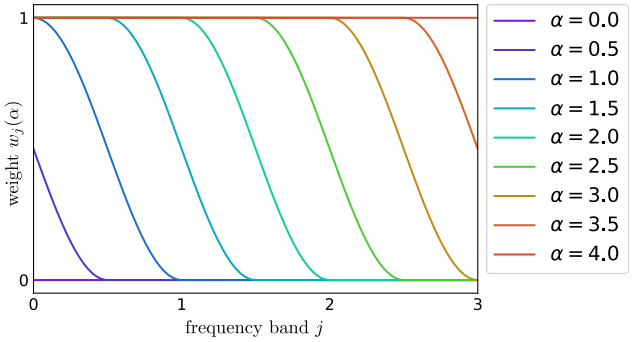


Figure 15: A visualization of the window function $w_j(\alpha)$ for the annealed positional encoding. We show an example with a maximum number of frequency bands of $m = 4$ where $j \in \{0, \ldots, m-1\}$. $\alpha = 0$ sets the weight of all frequency bands to zero leaving only the identity mapping, while an $\alpha = 4$ sets the weight of all frequency bands to one. Increasing the value of α is equivalent to sliding the window to the right across the frequency bands.

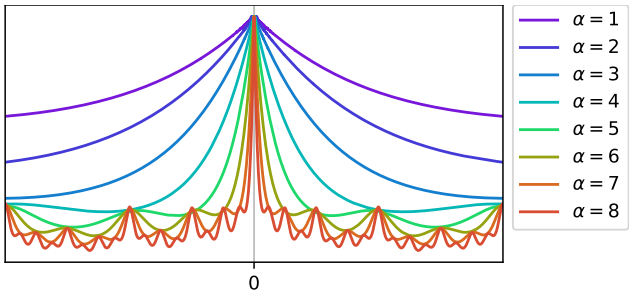


Figure 16: Visualizations of the neural-tangent kernel (NTK) [22] of our annealed positional encoding for different values of α. Our coarse-to-fine optimization scheme works by easing in the influence of each positional encoding frequency through a parameter α. This has the effect of shrinking the bandwidth of the NTK corresponding to the deformation MLP as α is increased, thereby allowing higher frequency deformations.

coordinate frame that is centered at the facemesh, with a standard orientation ($+y$ up, $+x$ right, $-z$ into the face), and with approximately metric units, by setting the scale so that the distance between the eyes matches the average interpupillary distance of 6 cm. Note that the 3D triangulation of facial landmarks is only correct if the subject is static, which is not guaranteed in our method, but in practice we observed that the triangulation result is sufficiently good to define the coordinate frame even when the subject rotates the head side-to-side. For the animal captures, we manually generate virtual camera paths.

References

[1] Kara-Ali Aliev, Artem Sevastopolsky, Maria Kolos, Dmitry Ulyanov, and Victor Lempitsky. Neural point-based graphics. *arXiv:1906.08240*, 2019. 2

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

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

[4] 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, 2020. 2

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

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

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

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

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

[10] Aljaž Božič, Pablo Palafox, Michael Zollhöfer, Angela Dai, Justus Thies, and Matthias Nießner. Neural non-rigid tracking, 2020. 2

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

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

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

[14] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation, 2017. 7, 10

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

[16] 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

[17] 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

[18] 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

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

[20] Richard Langton Gregory. The intelligent eye. 1970. 7

[21] 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

[22] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *NeurIPS*, 2018. 5, 10

[23] Chiyu "Max" Jiang, Jingwei Huang, Andrea Tagliasacchi, and Leonidas Guibas. ShapeFlow: Learnable deformations among 3d shapes, 2020. 2

[24] Abhishek Kar, Christian Häne, and Jitendra Malik. Learning a multi-view stereo machine. In *NIPS*, pages 365–376, 2017. 2

[25] 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

[26] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 8

[27] 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

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

[29] 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, 9

[30] 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

[31] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, et al. MediaPipe: A framework for building perception pipelines. *arXiv preprint arXiv:1906.08172*, 2019. 6, 10

[32] Ricardo Martin-Brualla, Rohit Pandey, Shuoran Yang, Pavel Pidlypenskyi, Jonathan Taylor, Julien Valentin, Sameh Khamis, Philip Davidson, Anastasia Tkach, Peter Lincoln, et al. LookinGood: Enhancing performance capture with real-time neural re-rendering. *SIGGRAPH Asia*, 2018. 2

[33] 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. *arXiv*, 2020. 2, 3, 10

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

[35] Moustafa Meshry, Dan B Goldman, Sameh Khamis, Hugues Hoppe, Rohit Pandey, Noah Snavely, and Ricardo Martin-Brualla. Neural rerendering in the wild. In *CVPR*, 2019.

2

[36] 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, 9

[37] 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

[38] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. HoloGAN: Unsupervised learning of 3d representations from natural images. *ICCV*, 2019. 2

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

[40] 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

[41] Keunhong Park, Arsalan Mousavian, Yu Xiang, and Dieter Fox. LatentFusion: End-to-end differentiable reconstruction and rendering for unseen object pose estimation. In *CVPR*, 2020. 2

[42] 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, 9

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

[44] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. *CVPR*, 2016. 6, 7, 10

[45] Johannes Lutz Schönberger, True Price, Torsten Sattler, Jan-Michael Frahm, and Marc Pollefeys. A vote-and-verify strategy for fast spatial verification in image retrieval. *ACCV*, 2016. 6

[46] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. *arXiv preprint arXiv:2007.02442*, 2020. 2

[47] 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

[48] 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

[49] Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhöfer. Deepvoxels: Learning persistent 3D feature embeddings. In *CVPR*, 2019. 2

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

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

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

[53] 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, 5

[54] A. Tewari, O. Fried, J. Thies, V. Sitzmann, S. Lombardi, K. Sunkavalli, R. Martin-Brualla, T. Simon, J. Saragih, M. Nießner, R. Pandey, S. Fanello, G. Wetzstein, J.-Y. Zhu, C. Theobalt, M. Agrawala, E. Shechtman, D. B Goldman, and M. Zollhöfer. State of the art on neural rendering. *Computer Graphics Forum*, 2020. 2

[55] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *ACM TOG*, 2019. 2

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

[57] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers*, 2003. 7

[58] 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

[59] Egor Zakharov, Aliaksandra Shysheya, Egor Burkov, and Victor Lempitsky. Few-shot adversarial learning of realistic neural talking head models. *ICCV*, 2019. 2

[60] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. NeRF++: Analyzing and improving neural radiance fields, 2020. 2

[61] 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

[62] 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

[63] 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

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