TECHNISCHE UNIVERSITÄT BERLIN

Deformable Neural Radiance Fields

NeRFs, W-NeRFs and Nerfies

Awsam Agbaria

Hot topics in Computer Vision

10.06.2025

TECHNISCHE UNIVERSITÄT BERLIN

Outline

TECHNISCHE UNIVERSITÄT BERLIN

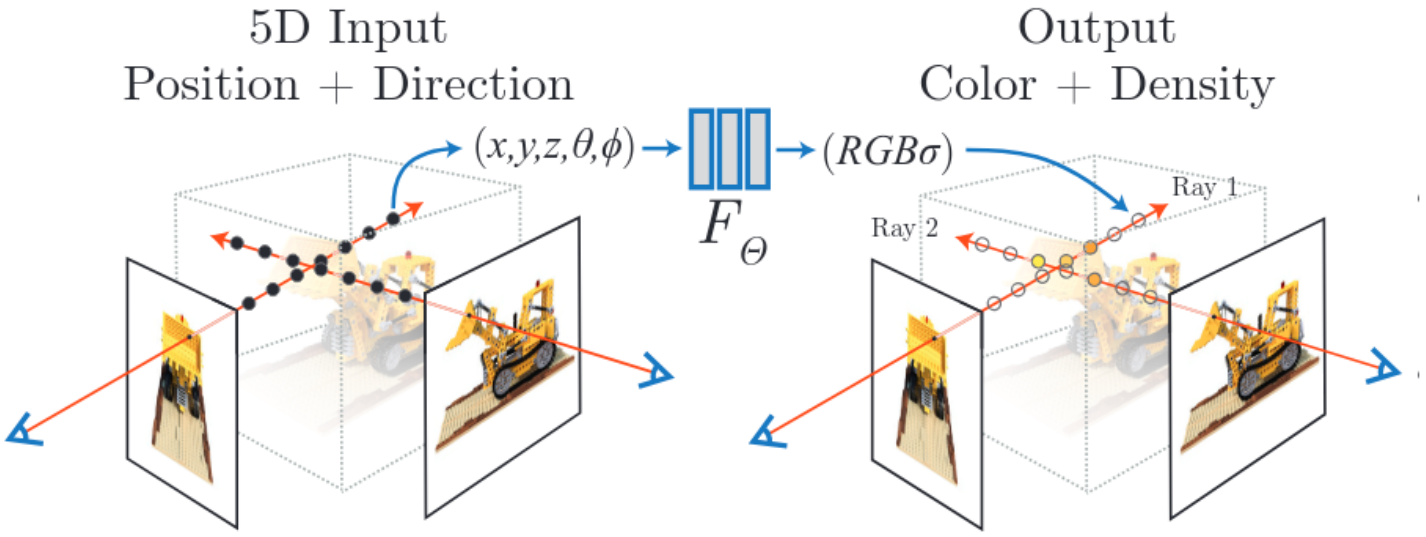
Problem formulation

Provided a set of images of a **static scene**, alongside their intrinsic and **varying** extrinsic characteristics:

- Can we extract information from different views of the same scene?
- Can we construct a scene from varying 2D views?
- Can we generate novel views?
- Can this process be learned by a neural network?

Neural Radiance Field

- A learned continuous implicit representation of a **static scene**'s density and color.



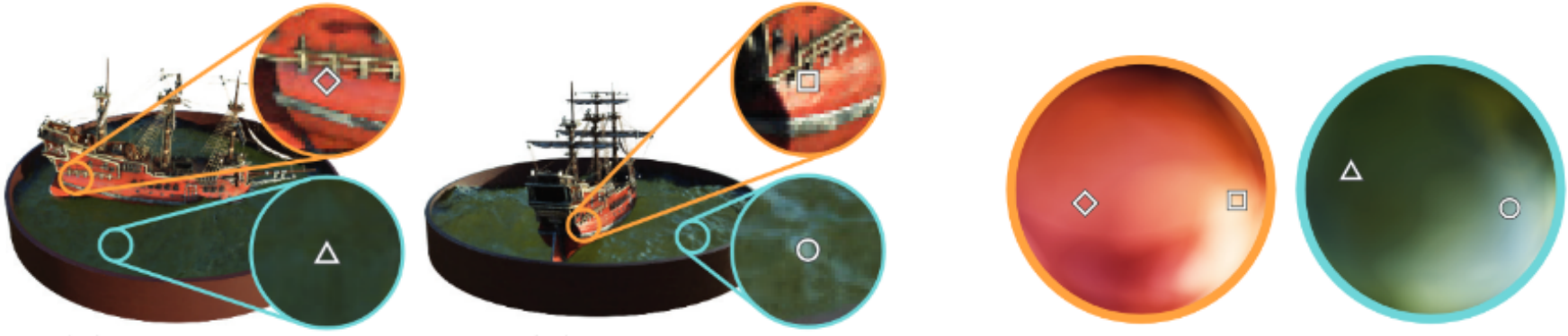
[3]

TECHNISCHE UNIVERSITÄT BERLIN

Volume Color

At every single point in the scene's volume we want to learn:

- The RGB true color at that coordinate (x,y,z).
- The reflectance of global illumination on that point from our current view.

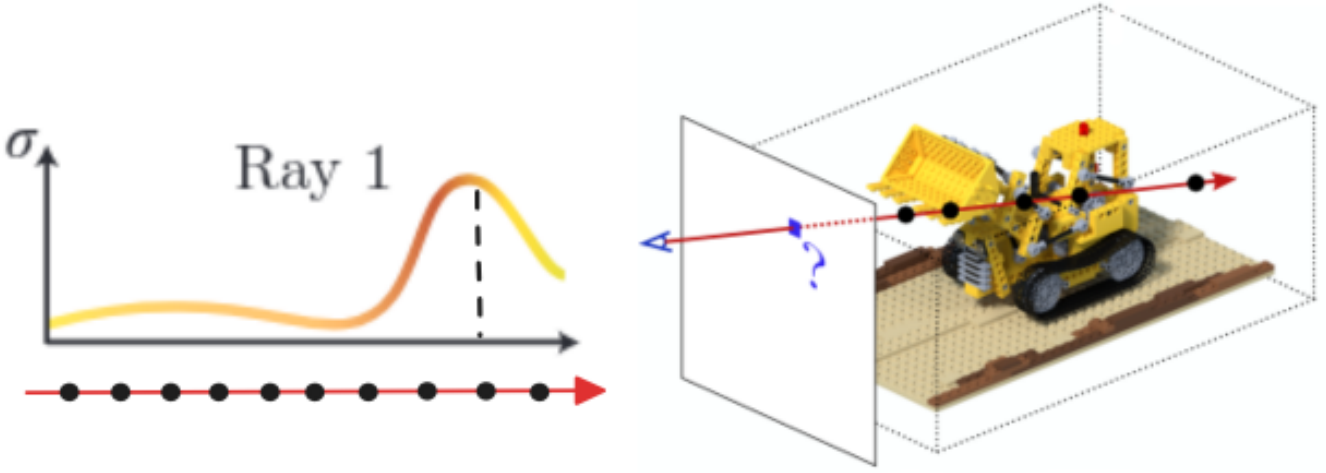


[3]

Volume Radiance Density

At every single point in the scene's volume we want to learn:

- A value [0,1] indicating the color radiance at that coordinate (x,y,z).
- Not orientation-dependent, must be multi-view consistent!

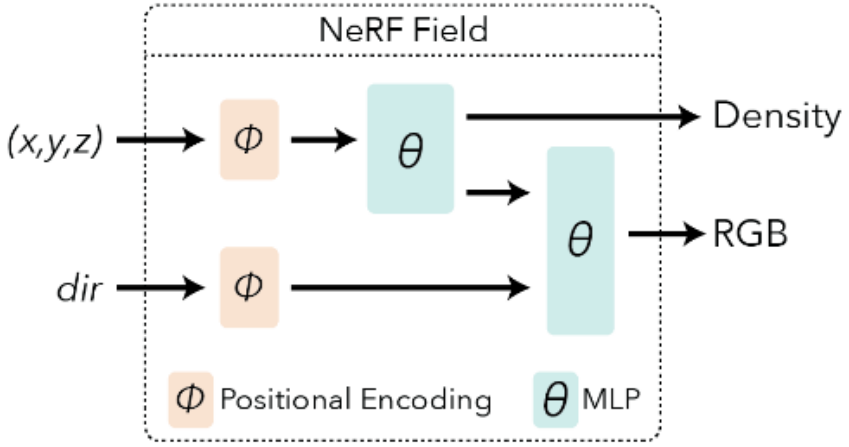


[3]

Learning the mapping function

A Neural network is then trained on this data to learn F_Θ.

- Neural networks (MLP) are universal function approximators.
- Unlike traditional deep learning, we do not care about generalization to the real world data distribution of similar objects, instead the neural network is trained to overfit the training data itself.



[3]

TECHNISCHE UNIVERSITÄT BERLIN

MLP biases

Through the findings of Rahaman et al. On the Spectral Bias of Neural Networks[6], it has been shown that:

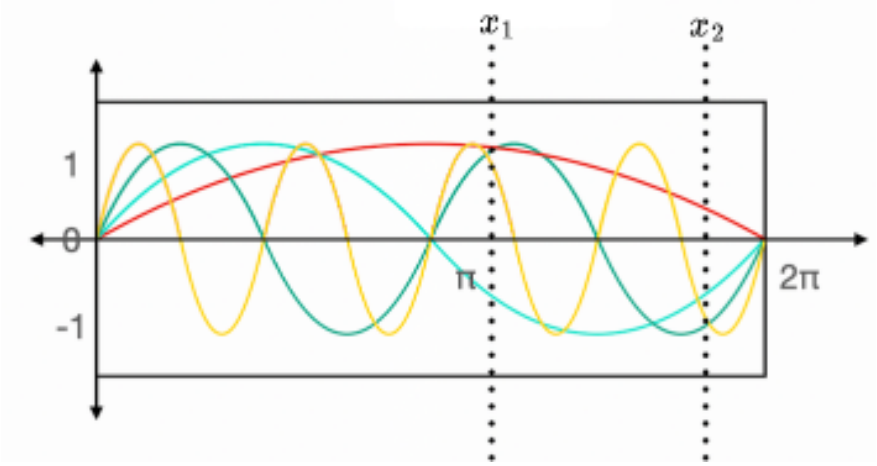
- Neural networks, particularly those with ReLU activations, tend to be biased towards fitting low-frequency patterns.
- The frequency spectrum of the underlying data manifold significantly influences learning bias of the neural network.

Expanding our data's frequency spectrum is essential for NeRFs in order to be able to model small intricacies.

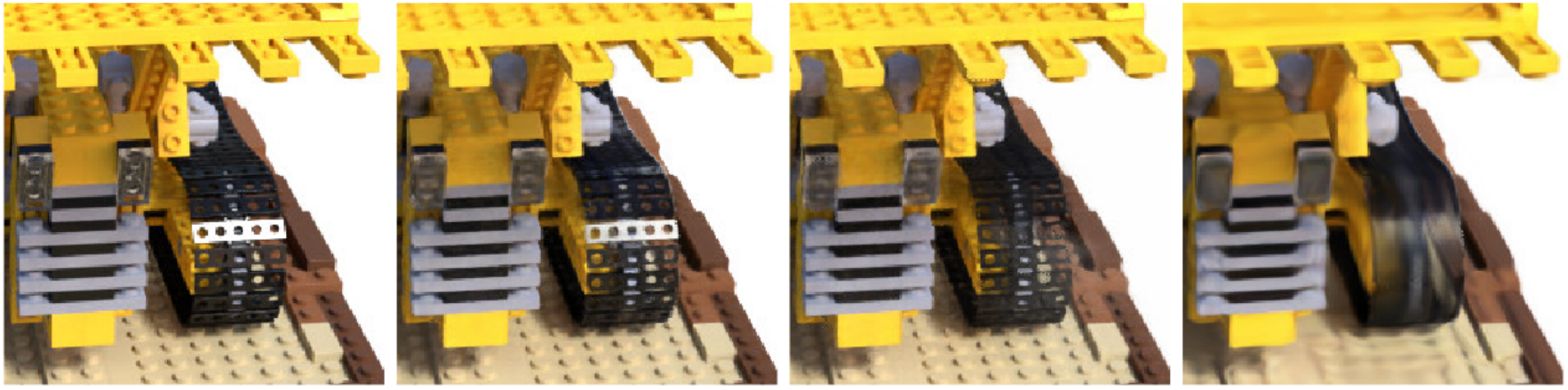
Positional Encoding

We map the features (x, y, z, θ, ϕ) into a higher-dimensional embedding via:

$$\Phi_L(x) = \begin{bmatrix} \sin(2^0 \pi x) \\ \cos(2^0 \pi x) \\ \sin(2^1 \pi x) \\ \vdots \\ \sin(2^{L-1} \pi x) \\ \cos(2^{L-1} \pi x) \end{bmatrix}_L$$



Results



Ground Truth | Complete Model | No View Dependence | No Positional Encoding

[3]

TECHNISCHE UNIVERSITÄT BERLIN

Limitations and Challenges

1. Unable to tackle Photometric variation.
2. Unable to handle Transient objects.
3. Not friendly to casual capturing environments (Many synchronized cameras required).
4. Often struggling to model very granular details like hair.
5. Unable to handle non-static deforming scenes and time varying scenes.

Photos in the wild

Nerfs are able to model completely static scenes because they rely on the assumption that **two intersecting rays from two different angles** should yield the same color (with specular reflection variations).

[2]

TECHNISCHE UNIVERSITÄT BERLIN

GLO: Generative Latent Optimization

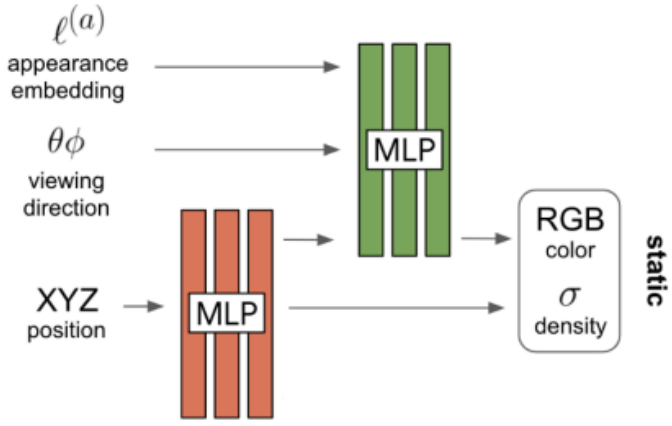
- To adapt to variable lighting each image i is assigned a corresponding real-valued photometric variations and illumination conditions embedding vector l_i^a of length n

- The image-independent radiance is now replaced with an image-dependent radiance i.e our MLP is now a mapping function conditioned on i-th embedding vector.

$$F_\Theta^i = MLP_\Theta(x, y, z, \phi, \theta, l_i^{(a)})$$

Photometric variations

- This vector is then optimized alongside the networks.
- Typically constrained to a low dimension to allow a smooth simple solution.
- Interpolated at inference.



[2]

[2]

TECHNISCHE UNIVERSITÄT BERLIN

Transient Objects

- Transient objects vary per image, similarly, image dependent embedding vector is introduced to capture those unique image features l_i^T
- We introduce a new MLP to model Transient object densities with an uncertainty output, the higher the uncertainty, the less reliable that transient object is meaning it is not part of the static geometry.

(a) Static (b) Transient (c) Composite (d) Image (e) Uncertainty

[2]

TECHNISCHE UNIVERSITÄT BERLIN

Limitations and Challenges

1. Unable to tackle Photometric variation.
2. Unable to handle Transient objects.
3. Not friendly to casual capturing environments (Many synchronized cameras required).
4. Often struggling to model very granular details like hair.
5. Unable to handle non-static deforming scenes and time varying scenes.

TECHNISCHE UNIVERSITÄT BERLIN

Motivation for Nerfies

The authors want to simplify the input process of NeRFs and make it more casual:

- Eliminate the need for multiple synchronized cameras.
- Allow variations in the illumination of the capture to account for auto-exposure and tone-mapping.
- Allow for simple and slow human-based deformations to take place.

TECHNISCHE UNIVERSITÄT BERLIN

Motivation for Nerfies

The authors want to simplify the input process of NeRFs and make it more casual:

- Eliminate the need for multiple synchronized cameras.
- Allow variations in the illumination of the capture to account for auto-exposure and tone-mapping.
- Allow for simple and slow human-based deformations to take place.

TECHNISCHE UNIVERSITÄT BERLIN

Human deformations

What type of deformations should we model?

1. Topology preserving or not?
2. Continuous or not?
3. Collapsing/Expanding to different dimensionality or not?
4. Rigid or non-rigid?
5. Smooth or not smooth?

TECHNISCHE UNIVERSITÄT BERLIN

Rigid Transformation

- Rigid transformations preserve the object's shape and size
- All distances and angles between all points before and after the transformation remain constant.
- Very simple mathematical formulation.
- Examples: Rotation, Translation (SE(3)).

Non-Rigid Transformation

- Rigid transformations **do not** preserve the object's shape and size, nor do they preserve angles and distances.
- Can be quite complex to learn, model and constrain (many mathematically valid solutions lead to the same result).
- Examples: Shearing, Scaling, Folding, Collapse, Tearing...

TECHNISCHE UNIVERSITÄT BERLIN

The Jacobian of a transformation

For a given Transformation T, the Jacobian J describes how infinitesimal changes happen to points in space during that transformation.

Theorem

if a Transformation T is **sufficiently continuously differentiable and smooth** at a point x, then locally around that point, the function behaves approximately linearly and can be approximated using its Jacobian matrix. Formally:

$$T(x + \delta x) \approx T(x) + J_T(x) \cdot \delta x$$

The Jacobian is thus also square invertible. $det(J) \neq 0$

TECHNISCHE UNIVERSITÄT BERLIN

Rigid Transformation

Given a rotation matrix R and a translation vector t, a rigid body transformation -and consequently its Jacobian- is defined as:

$$T(x) = Rx + t \qquad \rightarrow \qquad J_T(x) = R$$

Rigid transformations come with nice and convenient properties:

- J is a square and invertible matrix.
- J is orthonormal and preserves volume in space (det(J) = 1)
- All singular values of J are equal to 1.

TECHNISCHE UNIVERSITÄT BERLIN

Reminder: Singular Value Decomposition

Given any real $m \times n$ matrix A, its singular value decomposition is

$$A = U \Sigma V^T,$$

where

- $U \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n \times n}$ are orthogonal: $U^T U = I_m,\ V^T V = I_n$.
- $\Sigma \in \mathbb{R}^{m \times n}$ is diagonal (up to extra zero rows or columns), with nonnegative *singular values* $\sigma_1 \geq \sigma_2 \geq \cdots \geq 0$ on the diagonal.

SVD exists for *any* real (or complex) matrix and reveals rank, norms, and subspace structure.

TECHNISCHE UNIVERSITÄT BERLIN

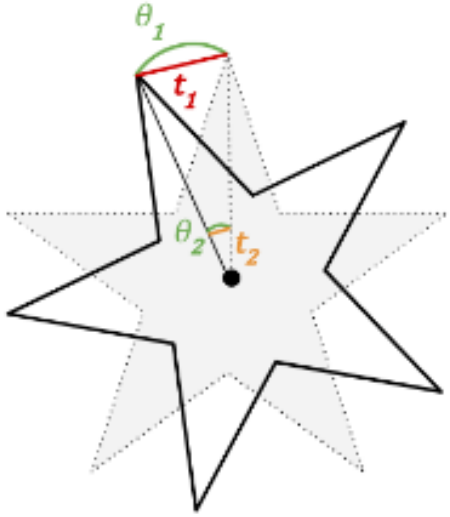
Modelling non-rigid deformations

Assuming we have a smooth, continuously differentiable, topology preserving transformation T, How can we model a simple, differentiable and constrained mapping for this transformation?

Naive Solution: Translation field
For every point x, the field would map it using a translation t(x) into x', therefore creating an origin and destination to every unique infinitesimal point in space and a translation at each key frame that it undergoes.

Modelling non-rigid deformations

Problem: This becomes super hard to optimize for large complex scenes and is inefficient for deforming objects.



[4]

TECHNISCHE UNIVERSITÄT BERLIN

Locally-Rigid Transformations

Locally rigid are **non-rigid** transformations that are rigid on infinitesimal local points.

$$T(x) = R(x) \cdot x + t(x)$$

Consequently its Jacobian is derived as follows:

$$J_T(x) = \frac{\partial}{\partial x}\big(R(x) \cdot x + t(x)\big) = \underbrace{R(x)}_{\text{Rotation}} + \underbrace{\big(\frac{\partial R(x)}{\partial x}\, x \; + \; \frac{\partial t(x)}{\partial x}\big)}_{\text{Spatial variations}}$$

Which does not preserve distances, nor are its singular values equal to 1!

TECHNISCHE UNIVERSITÄT BERLIN

Deformable NeRFs

Nerfies are Neural radiance fields that model small, smooth, topology preserving, non-rigidly deforming scenes given a set of casually captured images of the scene. This is achieved by decomposing the problem into two fields:

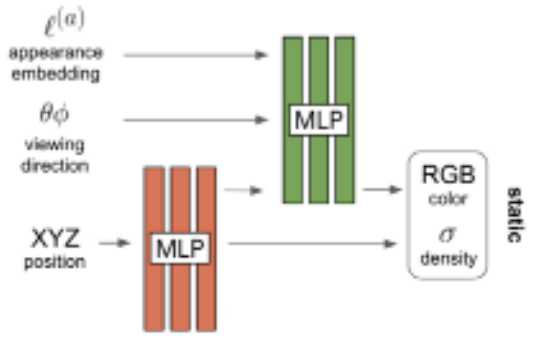
- **Canoncial:** Neural Radiance Field that models the geometry at the canonical rest frame.

- **Observation:** Neural Deformation Field that models the deformation of each point from the canonical frame into the observed frame.

Canonical NeRF

Standard NeRF that models the template volume of the scene at the canonical non-deformed state, with the assumption that there is photometric variations of capture since the input is typically done with one camera.

$$F_\Theta : (x, y, z, \theta, \phi, \ell_i^{(a)}) \rightarrow (\sigma, r, g, b)$$



[4]

TECHNISCHE UNIVERSITÄT BERLIN

Observation Deformation Field

For every frame i, we define an observation-to-canonical deformation that maps observation coordinates x to canonical coordinates $x^{'}$. We define this mapping as:

$$T : (x, \ell_i^{(d)}) \rightarrow x^{'}$$

Thus the Observation Neural Deformation Field G is defined as:

$$G(x, y, z, \theta, \phi, \ell_i^{(a)}, \ell_i^{(d)}) = F(T(x, y, z, \ell_i^{(d)}), \theta, \phi, \ell_i^{(a)})$$

TECHNISCHE UNIVERSITÄT BERLIN

Deformation map

We formulate the deformation map using a dense SE(3) field W that is **locally rigid**.

$$W : (x, \ell_i^{(d)}) \rightarrow (r, v)$$

such that (r,v) is the encoded screw axis of each local transformation.

Definition: Screw Axis

a geometric representation of a rigid body motion, combining a rotation about an axis $\frac{r}{\|r\|}$, instantaneous rotational motion $\|r\|$ and an instantaneous translation along that same axis v.

Screw Axis

We can use the 6d encoded vector to derive a 4x4 transformation matrix, we start by defining the skew symmetric matrix $[r]_x$ of r:

$$r = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix}, \quad [r]_\times a = r \times a \quad \rightarrow \quad [r]_x = \begin{bmatrix} 0, -r_3, r_2 \\ r_3, 0, -r_1 \\ -r_2, r_1, 0 \end{bmatrix}$$

This takes the encoded axis of rotation and gives us a matrix that rotates a point around that axis of rotation using the cross product.

TECHNISCHE UNIVERSITÄT BERLIN

Screw Axis - Rotation

We turn from a twist (instantaneous transformation) to rigid-body transformation using exponentials (continuous repetition):

$$R = e^{[r]_x}$$

Using the taylor expansion of matrix exponential and the cyclical behaviour of rotations we can decompose this into sines and cosines (Rodrigues' rotation formula):

$$e^A = \sum_{k=0}^{\inf} \frac{A^k}{k!} \quad \rightarrow \quad R = e^{[r]_\times} = I + \frac{\sin\theta}{\theta}[r]_\times + \frac{1-\cos\theta}{\theta^2}[r]_\times^2$$

TECHNISCHE UNIVERSITÄT BERLIN

Screw Axis - Translation

Similarly the overall translation p can be decomposed into the translation v and the path traced by the body along the rotation G.

$$p = Gv, \qquad p = v \int_0^1 R(\tau) d\tau$$

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

Putting it all together gives us the following map

$$W : x^{'} = Sx, \qquad \text{such that} \qquad S = \begin{bmatrix} R, p \\ 0, 1 \end{bmatrix}$$

Nerfies - Input Process

Data Recording:

- Input data: a 20s+ selfie video or a sequence of selfie photos
- It is assumed that the person is standing in front of a static background.
- The Video should include a diverse coverage of a 45 selfie cone.
- The authors used a second phone to generate a test dataset.

[4]

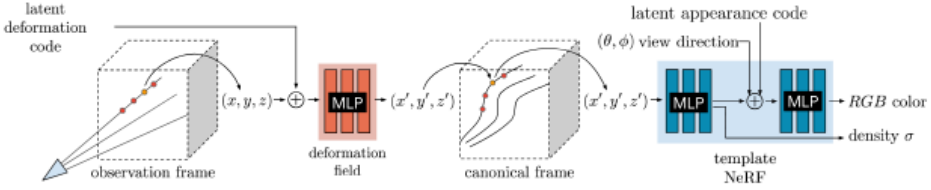
TECHNISCHE UNIVERSITÄT BERLIN

Nerfies - Input Process

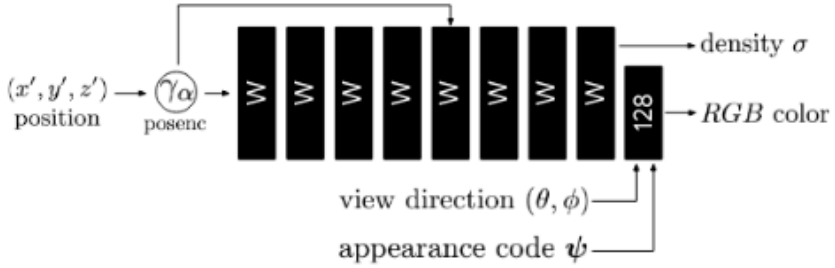
Data Preprocessing:

- Video is sampled to 5fps, and blurry captures are filtered and only about 600 captures are kept.
- COLMAP is used to infer the camera parameters for every frame (both intrinsic and extrinsic). However, this will produce inconsistent results in dynamic scenes.
- A foreground segmentation network is used in order to mask out the foreground for COLMAP.

Nerfies - Model

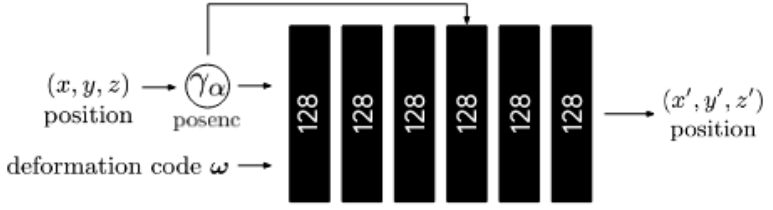


[4]



[4]

- Replacing ReLU with softplus.
- 6 hidden layers with 128 neurons.
- Skip connection for stable gradients.
- 80k Iterations with 6 frequency bands.
- latent variable dimension is 8.



[4]

TECHNISCHE UNIVERSITÄT BERLIN

Nerfies - Optimizations

Human movement is **mostly but not perfectly** rigid. The model still runs into ambiguities where multiple valid solutions occur, in which we typically favor the more rigid solution. This prior helps us constrain our problem further by introducing **Elastic Regularization** that enforces our solution to be **as rigid as possible:**

$$L_{elasitc}(x) = \| \log \Sigma - \log I \|_F^2 = \| \log \Sigma \|_F^2$$

- Note that our continuous formulation of continuously differentiable deformations allows us to directly compute J through automatic differentiation of the MLP.

Nerfies - Optimizations

[4]

TECHNISCHE UNIVERSITÄT BERLIN

Nerfies - Optimizations

Background Regularization: Solutions that include a moving background should not be considered, thus a penalty needs to be introduced:

$$L_{background} = \frac{1}{K} \sum_{k=1}^{K} \|T(x_k) - x_k\|_2$$

TECHNISCHE UNIVERSITÄT BERLIN

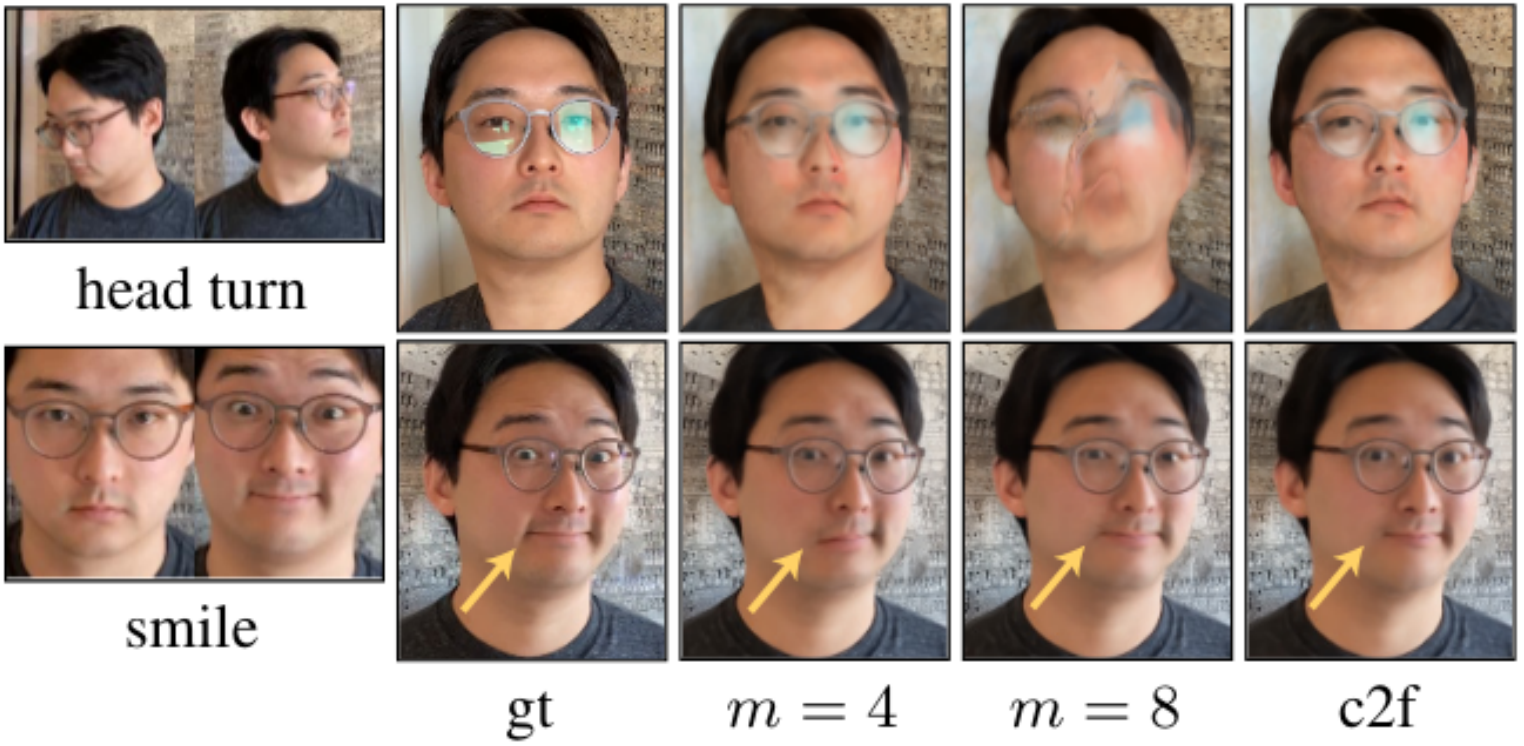
Nerfies - Optimizations

It was observed that when a combination of low and high frequency deformations were input, the model would often land in a local minima (Overfitting to one frequency).
To circumvent this, the authors proposed a coarse-to-fine frequency annealing strategy that smoothly shifts the frequency weight over many iterations:

$$w_i = \frac{(1 - \cos(\pi \cdot \mathit{clamp}(\alpha - j, 0, 1)))}{2}$$

- When $\alpha = 0$ only the first frequency band is enabled
- When $\alpha = m$ all frequency bands are enabled

Nerfies - Optimizations



[4]

TECHNISCHE UNIVERSITÄT BERLIN

Qualitative results

Figure 1: The model is able to portray small deformations and creases that would otherwise cause issues in static NeRF

[4]

TECHNISCHE UNIVERSITÄT BERLIN

Qualitative results

ground truth | rendered color | render w/o bg

TECHNISCHE UNIVERSITÄT BERLIN

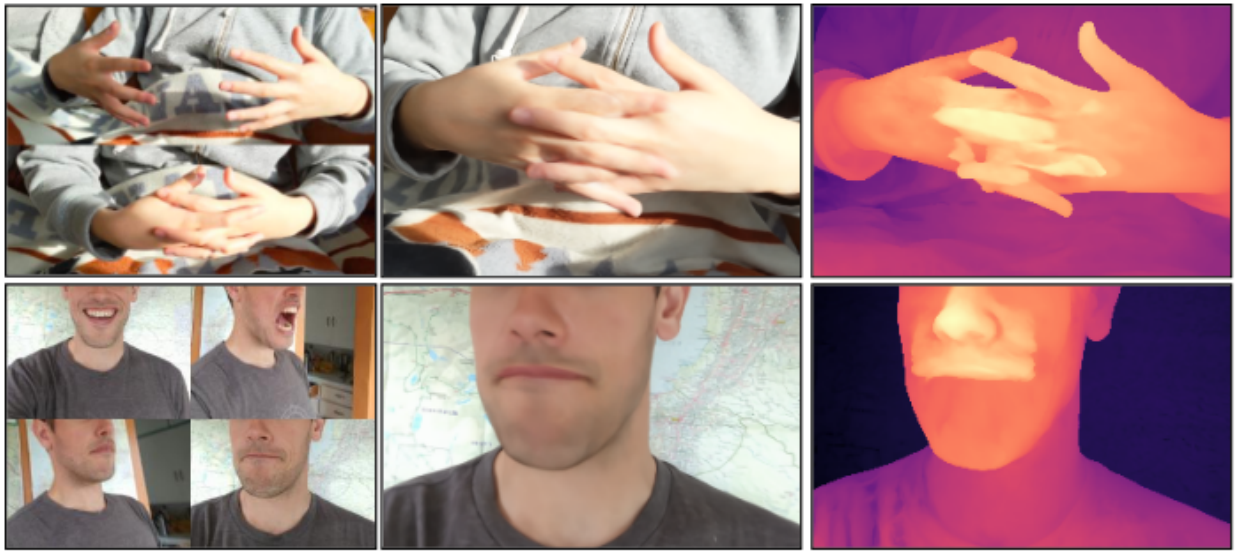
More results

- Quantitive results use two metrics, namely LPIPS and PSNR, however these metrics often favored blurry images over slight misalignment which is not a great representative of generation quality.
- Qualitative results are best seen through the paper and website:
- arXiv: https://arxiv.org/pdf/2011.12948
- Website: https://nerfies.github.io/

TECHNISCHE UNIVERSITÄT BERLIN

Limitations

- Due to our assumption of the transformation, the model struggles to represent topology changes:



example inputs rendered color rendered depth [4]

Limitations

- Rapid motion is quite hard to overfit on because the model filters out blurry frames and does not have enough data to overfit quick movements:

(a) input rgb

(b) rendered rgb

[4]

Limitations

- The hollow-face illusion is an optical illusion where a concave (pushed in) imprint of an object appears to be convex (pushed out) instead and following the viewers eye. This illusion is also a minor failure of this model that makes the geometry dynamic to fit the eye movement.

example inputs novel views for same deformation code [4]

TECHNISCHE UNIVERSITÄT BERLIN

Limitations

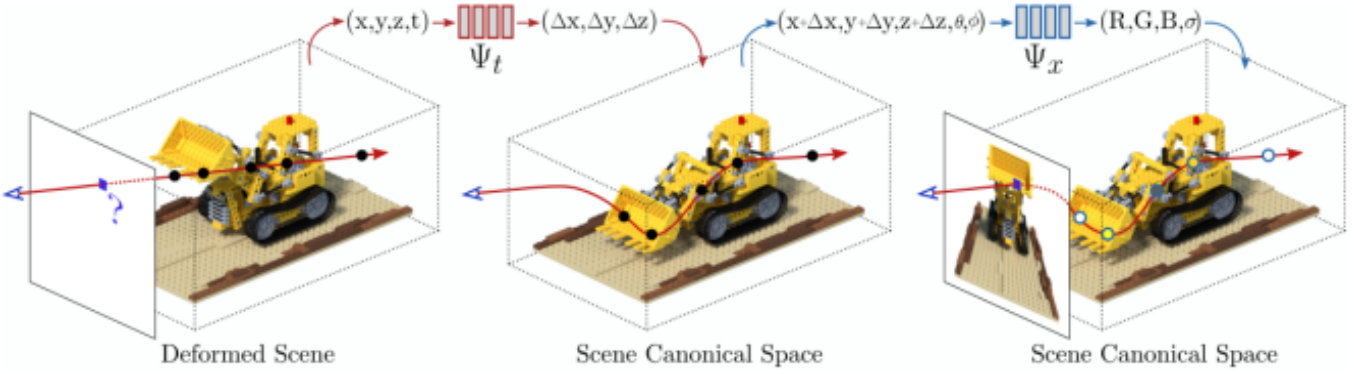
- The computational costs of this method deviate away from the ”Casual setting” it is meant to provide.
- Training on full resolution takes 3days on 8 V100 GPUs.
- Training on minimal resolution takes 17h on 8 V100 GPUs.
- A step in the right direction nonetheless.

TECHNISCHE UNIVERSITÄT BERLIN

Thank you for listening

D-NeRFs

Very similar Observation-to-Canonical formulation of the problem, however the deformation field in D-NeRF is time variant instead of frame variant. This method lacks further optimization that Nerfies has introduces to tackle real scenes.



[5]

TECHNISCHE UNIVERSITÄT BERLIN

Neural Scene Flow Fields

- Similarly to Nerfies, NSFF uses the Observation-to-Canonical formulation of the problem.
- Density and color are time variant.
- They introduce a ”flow” mechanism that conditions the current view on its previous+next view, such that the deformation is smooth along the frames. The flow represents a 3D offset vectors that point to the position in the previous/next frame.
- Introduce Occlusion weights that down-weight ambiguous occluded part to not affect the loss.

$$F_{\Theta}^{dyn}(x, d, t) = (c_t, \sigma_t, f_{t\to t+1}, f_{t\to t-1}, w_{t\to t+1}, w_{t\to t-1})$$

[1]

TECHNISCHE UNIVERSITÄT BERLIN

References I


Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang.
Neural scene flow fields for space-time view synthesis of dynamic scenes, 2021.


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, 2021.


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, 2020.


Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Steven M. Seitz, and Ricardo Martin-Brualla.
Nerfies: Deformable neural radiance fields, 2021.


Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer.
D-nerf: Neural radiance fields for dynamic scenes, 2020.

References II

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred A. Hamprecht, Yoshua Bengio, and Aaron Courville.
On the spectral bias of neural networks, 2019.