# Efficient Non-Rigid Neural Radiance Fields for Virtual Reality Video Conferencing

Candidate no. 1067959

University of Oxford

A thesis submitted for the degree of

*Master of Science in Advanced Computer Science*

August 29, 2023

# Table of Contents

# List of Figures

# List of Tables

## Acknowledgements[1]

Thank you to my first supervisor, for the exciting discussions and your honest thoughts, as well as your guidance and support.

Thank you to my second supervisor, for helping me focus on the core problem and not letting me get side-tracked by other interesting questions, and for your technical guidance on the subject.

Thank you to my third supervisor, for helping me shape my initial project and directing me towards the right people.

Thank you to the scholarship organization, who contributed towards the funding of my studies here.

Thank you to the student, who allowed me to use their face for the data.

Most importantly, thank you to my friends and family, for your endless and loving support in helping me navigate the ups and downs.

[1] Names omitted for anonymity purposes.

## Abstract

We present an efficient novel-pose and view synthesis model that can be used in downstream tasks such as virtual reality video conferencing systems. The system uses input from a standard webcam, and then generates different perspectives of the person in front of the camera. This enables the creation of a virtual round table with photo-realistic human avatars with minimal hardware requirements. While prior work on non-rigid scenes primarily deals with fixed-length videos, we adapt the architecture of neural radiance fields to deal with previously unseen facial expressions in video streams. To achieve the required real-time performance, we propose a simple preprocessing stage during training and inference which relies on existing priors to optimize ray sampling. By isolating the face region and using the head as a frame of reference, we reduce motion, allowing us to perform ray marching more efficiently. We compare our results against existing methods, as well as very recent advances from August.

# 1 | Introduction

> The most important thing in communication is hearing what isn't said.
>
> *Peter Drucker*

The human species is a social one; we live in communities, help each other when in need, develop and maintain lifelong friendships or romances. At the same time, as a species that has settlements in every continent of the world, it is unsurprising that our attempts at long-distance communication go far back. With the invention of the first electric telegraph in the 18th and 19th century [1], a milestone was reached.

While the transmission of discrete electrical signals was relatively straightforward, the encoding and decoding of sound into/from electrical signals was its own challenge. A few years later, on March 10, 1876, Alexander Graham Bell then made the first telephone call [2]. It was not much later that there was was in interest in not just hearing the person but also seeing them. However, the cameras for live video transmission did not exist yet.

Shortly after the first video cameras became commercially available, AT&T invented the first two-way video conferencing system in 1931, connecting two offices, but it wasn't until 51 years later that Compression Labs released the first commercial group video conferencing system; next to upfront investment costs of $250,000 for hardware that took up an entire room, calls cost $1,000 / hour [3, 4]. Today, especially since the COVID-19 pandemic, we take

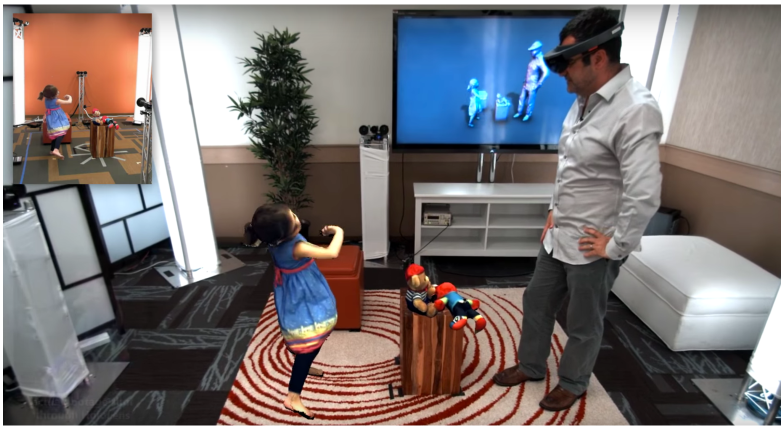

Figure 1.1: Holoportation: the augmented reality system brings two separate people into the same physical room. The system consists of eight carefully calibrated RGB-D cameras. Credits: Microsoft

video conferencing for granted. High-speed internet connectivity allow us to communicate with close ones or business partners - (often) for free, in color and in 4k resolution.

Now, what could long-distance communication look like 5 years from now? Communication is not just about words, nor is it just about our facial expressions. It includes our posture, whether we look a person in the eye or not, and whom we give our attention in what way[1]. A first idea was given by Microsoft through the Holoportation project [5] by connecting a parent and their daughter across two rooms in augmented reality (see figure 1.1).

## 1.1 Motivation and Relevance

In order to move away from a flat grid-aligned representation of faces in a video call, and towards a more immersive representation, we need means of

[1] Looking further down the line, it may also include the choice of cologne or perfume, the smell of a freshly cooked meal, a comforting hug for friends or family, or a hand shake with a business partner. These types of sensory transmissions are unfortunately out of scope for this project, but paint a picture of what digital communication could look like.

digitally capturing not just an image but the geometry, color and lighting of a person and scene. This procedure needs to run in real-time, with low latency and, in order to be adapted by wider audiences, require minimal additional or specialized hardware.

Scanning and reconstructing 3D models of our environment has been a long-standing area of interest in the Computer Graphics domain, and still an active area of research due to its relevance for academia and industry. Applications for accurate digitized assets of the real-world are numerous, and include assets for video games and movies, architecture, agriculture and farming, scene reconstructions for forensic sciences, faster and easier access to archaeological sights or objects, as well as threat detection and path planning for safer autonomous driving. A branch of research deals specifically with the realistic reconstruction of human avatars - be it for video games, or future applications in video conferencing.

Realism in real-time settings is still an active area of research and suffers from many issues. Using classical computer graphics methods, modelling realistic hair is a challenge due to the high-level of geometric detail required, even without simulating hair flow. Next, the material properties of the skin need to be captured accurately, and will differ from person to person. While classical real-time computer graphics has advanced tremendously in this regard as seen by the realism in current video games in figure 1.2, they are also hand-crafted characters that can each take weeks to create[2]. Furthermore, they can still be recognized as digital models - every so small fine detail needs to be manually implemented, and the scope of physically accurate rendering is restricted by computational capacity.

[2] Based on a character creation tutorial length [6]. May differ for AAA video games.

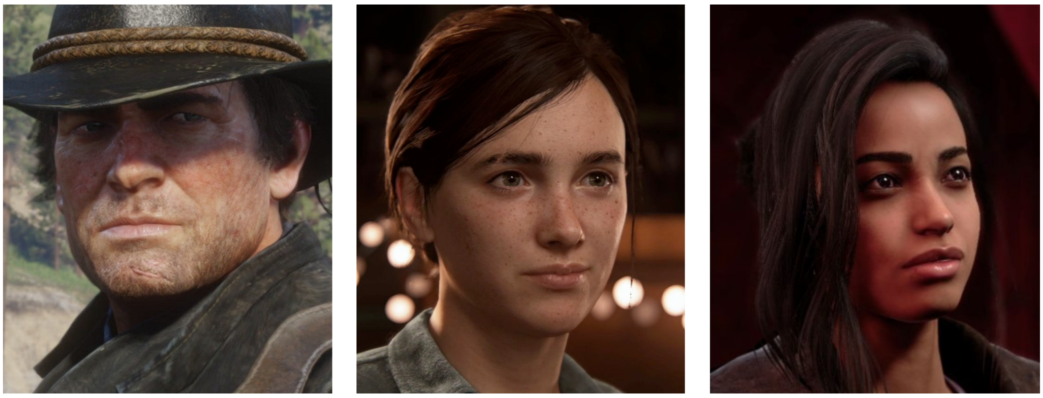

Figure 1.2: Realistic characters in current video games are carefully crafted 3D models. From left to right: Red Dead Redemption 2 (2018), The Last of Us Part II (2020), Forspoken (2023)

Due to these challenges, existing virtual reality conferencing software, such as MeetInVR, Meta's Horizon Workrooms and Microsoft Mesh only use simple 3D avatars instead of 3D scans and do not aim for photo-realism [7–9].

In 2020, a significant step towards neural rendering was made with the introduction of neural radiance fields (short: NeRF). The problem of photo-realism of (reconstructed) digital assets, including manually editing material and lighting parameters, was replaced by reformulating the problem, solving it top-down rather than bottom-up by having a neural network model these parameters implicitly.

Performance improvements in this field have motivated the use of neural rendering techniques for 3D/VR video conferencing. Our research aims to alleviate some of the remaining issues, with the long-term target of achieving photo-realistic mixed-reality social interaction from a monocular video alone.

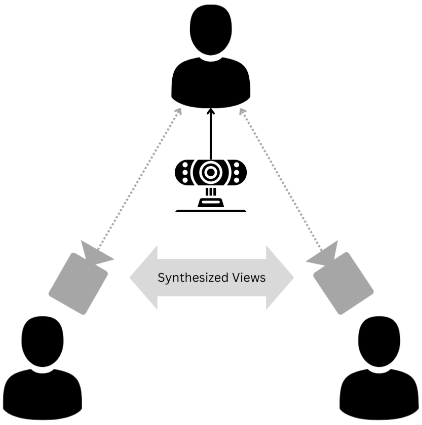

Figure 1.3: Creating a virtual round table: the system synthesizes different views of a person's face, from a single front-facing camera.

## 1.2 Guiding Questions and Methodology

The primary goal of this study is to develop an **efficient rendering method and inference pipeline** in order to leverage recent advances in neural radiance fields for virtual reality video conferencing. We assume a standard user setup with a monocular webcam with the objective of synthesizing a different view for each participant in the video call as seen in figure 1.3.

In order to achieve real-time performance during inference without sacrificing quality, we employ two different methods during training and inference. During training - consisting of a short video sequence of an individual speaking - we use a pre-trained deep residual network to accurately estimate the head pose and facial expressions, supported by a 3D morphable face model. At this point, we propose a change of the frame of reference to reduce the magnitude of deformations, providing a basis for improved ray-sampling. In the training stage, we construct a density field that indicates the maximum occupancy of a specific individual in space, across all expressions.

After obtaining a detailed neural avatar, we use a pre-trained light-weight model to estimate a slightly less accurate head pose and facial expression. These inaccuracies do not matter during inference, as participants in a video call would not see the ground truth image - the sole purpose of the webcam now is to identify the pose and expression. We use the previously constructed density field to efficiently sample points in spaces that are actually occupied. The solution is mostly implemented in PyTorch.

The second guiding question deals with the observation in current literature and our work that certain deformations cause visual artifacts, including opening and closing the mouth or eyes. We formalize these challenges by drawing a link to topology, and discuss the expressiveness of a potential solution to the problem.

## 1.3 Thesis Structure

This thesis has six chapters - the first, this one, contextualizes the problem and provides a high-level methodological overview. The literature review in chapter 2 then goes into more extensive detail of the research field, spanning classical approaches from 2015 to the state of the art of Neural Radiance Fields in August 2023.

In chapter 3, we introduce the reader to the required background knowledge on computer graphics and NeRFs - covering camera models, rigid transformations, volumetric rendering and deformations.

Our contributions, discussed in chapter 4, are threefold:

- We analyze the performance bottlenecks in rigid and non-rigid settings, which motivates the emphasis on efficient point sampling and rendering

- With the help of priors, we introduce a training pipeline which normalizes the head pose.

First, we motivate the focus of this thesis by analyzing the performance bottlenecks in rigid and non-rigid settings, to then address them in section 4.2 where we introduce priors into the training pipeline. Thirdly, we propose improvements to the pipeline following inaccuracies in the automated head-pose estimation.

We conclude the thesis with an outlook of promising remaining research questions in chapter5, as well as as a critical evaluation of our studies and its limitations in the final chapter 6. Here, we also elaborate on the societal impact of the technology, with the hope of appealing to responsible AI and policy researchers.

# 2 | Literature Review

In general, the literature distinguishes between rigid and non-rigid reconstruction, i.e. whether the captured scene is static or dynamic. The research can be approximately categorized into classical and learning-based approaches. Classical approaches typically rely on structured light, RGB-D cameras [10], stereo cameras, structure-from-motion or multi-view camera rigs. One of the notable works for mixed-reality social interaction using classical volumetric fusion from RGB-D observations is Holoportation [5].

Since 3D scanning is often under-constrained, especially in non-rigid settings, learning-based approaches have a lot of potential. Some early work deals with reconstructing a rigid 3D-voxel object from a single RGB image using a trained recurrent neural network, such as 3D-R2N2 [11].

A lot of attention has been given to Neural Radiance Fields [12], which don't so much deal with 3D reconstruction, but rather with synthesizing images of an object from novel perspectives. An MLP $\theta$ is trained to return the volume density and color of a 3D point in space, conditioned on the viewing angle.

However, with training for a single scene taking up to days in the original publication, significant work has gone into optimizing the fields. By splitting the NeRF into a view direction and position MLP, [13] can efficiently cache many computations, achieving inference speeds of up to 200 FPS on static scenes. While early methods rely on Fourier features as introduced by [14], subsequent works aim at improving efficiency by building sparse feature grids. Plenoxels [15] achieves improvements by entirely eliminating neural components, and uses spherical harmonics in a sparse 3D grid that can be optimized through gradient methods, with training times in the order of

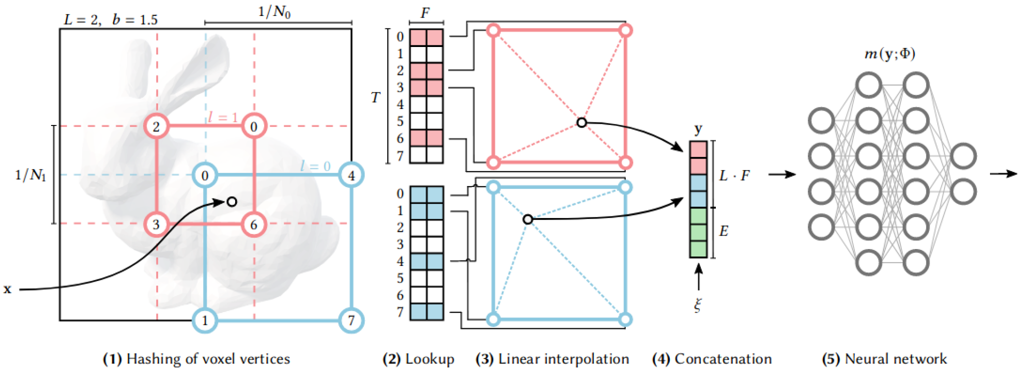

Figure 2.1: Multi-resolution hash encoding as implemented in Instant-NGP [16]. Image from original publication.

magnitude of minutes instead of days. The next milestone in terms of performance was reached with Instant-NGP [16] which achieved training times in seconds by using density grids for efficient ray sampling and multi-resolution hash encoding which stores trainable features to allow for a lighter network architecture (figure 2.1). Using a hash table instead of a grid furthermore reduced the memory requirements.

Using a hierarchical representation and learning compressed feature grids, [17] achieves variable bitrates for progressive streaming of data, allowing to adapt the amount of data accessed to render an image based on available bandwidth. NeRFLight [18], another feature grid approach, splits a scene into different regions with different decoders, but reusing the same features. This allows them to achieve good performance with low memory requirements. Most of the research is done on high-end consumer GPUs; BakedSDF [19] therefore aims at inference on commodity hardware using a method involving classical 3D meshes combined with a view-dependent appearance model to improve the performance. Other improvements have focused on disentangling the lighting from the scene, allowing to train on images with different lighting conditions, as well as render the image under different lights

[20].

So far, all of these methods assumed calibrated images from multiple viewpoints of a static scene, that is, none of the objects are changing or deforming during captures. Not soon after the NeRF [12] publication, a separate branch of research therefore starts dealing with the case of *non-rigid* scenes, typically assuming fixed-length videos as input. Their objective is to synthesize novel viewpoints of the video for any point in time. To do so, most publications use variations of a deformation network introduced in D-NeRF [21], which predicts the difference of any point $x$ at time $t$ to the initial frame. NeRFies [22] improves robustness of the deformation by using elastic regularization of the field. To address discontinuities in deformation fields, HyperNeRF [23] increases the dimensionality of the space, taking inspiration from level set methods. To constrain rigid regions from deformations, [24] uses an additional rigidity network to weight the deformation.

There is less research on performance improvements for non-rigid scenes as for rigid ones, partially due to fewer known constraints that can be leveraged. TiNeuVox [25] greatly improves the training time by introducing coarse-to-fine time-aware voxel features, due to faster convergence. [26] in turn leverages results from [16], adapting the density field (for efficient point sampling) to non-rigid settings by taking the maximum occupancy across all time steps. The performance gains depend on the extent of the deformations.

While these non-rigid methods focus on fixed-length videos, i.e. conditioning the deformations on time, there has been an interest in conditioning the deformations on other variables such as human poses or facial expressions - thus extending the applications beyond novel-view synthesis in videos. One of the first controllable facial avatars was introduced by [27], which makes minor modifications to the original NeRF [12] architecture and does not use a deformation network, but directly conditions the main network on the expression. Both [28, 29] use human pose estimation to estimate the deformations and learn a canonical representation of a person. [29] specifically targets novel pose synthesis, but neither of the approaches aim for real-time performance. RigNeRF [30] works on pose and expression control with novel view synthesis using a 3DMM.

**Recent Developments.** The research field on neural radiance fields is developing quickly, and a lot of the research has advanced since the original research proposal for this dissertation. While some of the new developments have been taken into consideration, others have been too recent to update this research. [31], published in SIGGRAPH 2023 on August 6, replaces the NeRF with a Triplane encoder which learns features on orthogonal planes for more efficient inference. The authors do not require a separate training stage for each individual, and achieve real-time performance for novel-view and pose synthesis. In this publication, results are also compared with HeadNeRF [32] which uses NeRFs as proxies rather than for an underlying geometric structure. HeadNeRF achieves similar results, but has a very different methodology.

# 3 | Preliminaries

One of the reasons why neural radiance fields perform so well for novel view synthesis is because they embed constraints from traditional computer graphics into neural networks. And while the architectures of the networks often remain very simple, a good understanding of camera models, transformations and rendering concepts is required when working with NeRFs. We introduce these fundamental concepts of computer graphics before going into the exact functioning of the basic model architecture. We summarize some of the core computer graphics concepts from [33] here, while adopting some simplified notations.

## 3.1 Camera Model and Transformations

The representation of virtual environments on a 2D screen is crucial to computer graphics, and takes inspiration from the pinhole camera model. In this model, the light rays enter a closed box through a small hole, resulting in a flipped image on the back side of the box as seen in figure 3.1.

The distance between the pinhole and the film is typically referred to as the

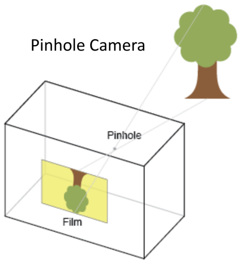

Figure 3.1: The pinhole camera model illustrates how an scene can be projected onto a screen. Image credits: TUM

focal length $f$ of the camera [1], and it corresponds to the length of the principal ray - the ray which is orthogonal to the film (starting at the principal point) and passes through the pinhole. The film (or image) plane has its origin in the center of the image, whereas digital images traditionally have their origin in the lower left corner. A translation therefore has to be applied when projecting points to a digital image.

While in analog devices the film is placed *behind* the pinhole due to physical limitations, in a virtual camera we can easily place the film *before* the pinhole, resulting in an upright (but otherwise identical) image, as seen in figure **??**.

In practice, we do not know the exact measurements of a camera. During production of the camera there can be minor aberrations so that one cannot fully rely on specification sheets, and would need to take the camera apart in order to perform exact measurements. Since that is unfeasible, parameters are determined through a camera calibration process, usually by taking images of a checkerboard from different angles. When calibrating a camera, we cannot determine the exact focal length, nor the sensor width and height. However, we can determine $f_x = f \cdot s_x$ and $f_y = f \cdot s_y$ where $s_x, s_y$ are the sensor width and height respectively. The sensor dimensions are measured in $[pixels/mm]$. Furthermore, the image plane of a camera might be skewed, and is captured by the skew parameter $\gamma$ during the calibration process. For most modern cameras, this value will be negligibly zero.

### 3.1.1 World, Camera and Pixel Space

When working with coordinates in a virtual environment, there are three commonly used frames of references. The first is the world space - this is the

[1] The term focal length is a bit of a misconception, as we do not really have a focal point in the pinhole camera model due to the lack of lenses.

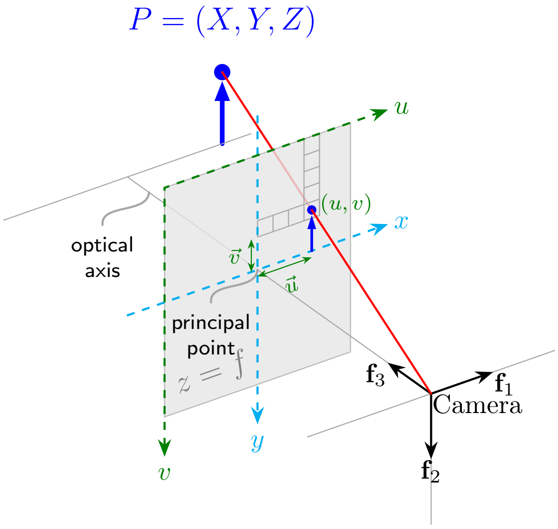

Figure 3.2: Camera frame of reference and the corresponding screen projection. Modified from [34]

global coordinate system, often using the earth as a frame of reference. We denote the basis of this vector space as $\mathscr{B}_w = \{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$. The second is the camera space with basis $\mathscr{B}_c = \{\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3\}$ - here, the camera is the center and points are described in relation to the camera. Finally, the pixel space is a projection of points in the camera space on to the hyperplane spanned by $\{\mathbf{f}_1, \mathbf{f}_2\}$.

If $P = x\mathbf{f}_1 + y\mathbf{f}_2 + z\mathbf{f}_3$ is a point in space, then $z$ is the distance of $P$ from the camera, by the choice of the basis $\mathscr{B}_c$. The principal point which lies on the film has the coordinates $(0, 0, f)$, with $f$ being the focal length.

Switching between the two coordinate systems is simply achieved through a change of basis. In computer graphics, the change of basis matrix $E$ from $\mathscr{B}_w$ to $\mathscr{B}_c$ is referred to as the camera extrinsic matrix and determines the position and orientation of the camera. If $P_w = [x, y, z, 1]^T$ are the homogeneous coordinates of a point in the world space, $P_c = EP_w$ are the coordinates of $P_w$ in camera space. We discuss the construction of $E$ in section 3.1.2.

The matrix to project a point $P_c$ from the camera space to the screen is called the *camera intrinsic matrix* $K$, and applying it gives us the pixel coordinates $(u, v)$:

$$K \begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} = \begin{pmatrix} f_x & \gamma & m_x \\ 0 & f_y & m_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} = z_c \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}. \tag{3.1}$$

Similarly, adding a final zero column in $K$, if $E$ is the camera extrinsic matrix and $P = x\mathbf{e}_1 + y\mathbf{e}_2 + z\mathbf{e}_3 = x_c\mathbf{f}_1 + y_c\mathbf{f}_2 + z_c\mathbf{f}_3$, then the pixel coordinates of $P$ are

$$KEP = K \begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} = z_c \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \tag{3.2}$$

Note that this is a projection, and that multiple points are mapped to the same coordinates $(u, v)$. This line is referred to as a *ray* - an important concept for rendering, since all the points[2] along a ray will contribute towards the color of one specific pixel.

### 3.1.2 Rigid Transformations

The rotation of a vector in 3D space can be expressed as the rotation around each of the basis vectors. By convention in computer graphics [33], we first

[2] early termination aside

rotate a point along the $z$, then the $y$ and finally the $x$ axis:

$$R = R_x(\gamma) \cdot R_y(\beta) \cdot R_z(\alpha) \tag{3.3}$$

where $R_x, R_y, R_z \in \mathbb{R}^{3\times3}$ are the corresponding rotation matrices. The vector $v \in \mathbb{R}^3$ when rotated is given by $v' = Rv$, and when translated by $t \in \mathbb{R}^3$, it is $v' = v + t$. Both the rotation $R$ and translation $t$ can be expressed in a single operation when extending $v = [x, y, z]^T$ to homogeneous coordinates:

$$v' = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} v = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x & y & z & 1 \end{bmatrix}^T \tag{3.4}$$

## 3.2 3D-Morphable Face Models

Since human faces have common features, such as eyes, a nose and a mouth, considerable research effort has gone into developing mesh priors of human faces. As not every face is the same, these priors can be conditioned on a latent space for head shape, as well as expression and texture. The Basel Face Model (BFM) [35] for example relies on PCA bases. BFM consists of two components $\mathbf{S}$ and $\mathbf{T}$, corresponding to the shape and texture of a head:

$$\begin{aligned} \mathbf{S} &= \mathbf{S}(\alpha, \beta) = \bar{\mathbf{S}} + \mathbf{B}_{id}\alpha + \mathbf{B}_{exp}\beta \\ \mathbf{T} &= \mathbf{T}(\delta) = \bar{\mathbf{T}} + \mathbf{B}_t\delta \end{aligned}$$

where $\bar{\mathbf{S}}, \bar{\mathbf{T}}$ are the average face shape and textures, and $\mathbf{B}_{id}, \mathbf{B}_{exp}$ and $\mathbf{B}_t$ are the PCA bases of the face identity, expression and texture, controlled by the coefficients $\alpha \in \mathbb{R}^{80}, \beta \in \mathbb{R}^{64}, \delta \in \mathbb{R}^{80}$.

### 3.2.1 Reconstruction from Image

In practice, it can be useful to obtain the parameters of the 3D-morphable face model of a specific individual from an image. This problem is addressed by several works: In [36], a Resnet-50 architecture is used to predict identity, expression, texture, head pose and lighting parameters.

## 3.3 Neural Radiance Fields

Neural radiance fields, short NeRF, are an approach to digitally capture real-world scenes, similar to 3D reconstruction, from images alone. However, instead of computing a 3D polygonal mesh and texture which can then be rendered, Neural Radiance Fields learn an implicit representation of the scene, from which an image is computed.

The implicit representation models scene properties for any coordinate in the space, within the observable bounds. In its most basic form, the function models two properties: the volume density as well as the color at a given point. Unlike physical density, high density values in the context of volumetric rendering and computer graphics correspond to solid objects, while a low density indicates empty space[3]. Sometimes, the term *opacity* is used instead [37][4].

From this implicit representation that returns color and density values for any point in space, an image can be computed. For each pixel in the screen,

[3] For the physicists among the readers, we consider air to be empty space.

[4] In classical volume rendering, opacity is more common, whereas in the context of Neural Radiance Fields, volume density is used, adapting conventions set out by [12]

a corresponding ray is cast into the scene along which we integrate over the product of density and color values to compute the final pixel color.

A NeRF can be trained from a dataset of images with known camera poses. Each batch consists of randomly selected rays from the pixels of the cameras in the training data. The integral is approximated by sampling points along each ray, and for each point, the neural radiance field computes corresponding color and density values. The loss function is a total squared error between the rendered and actual pixel value from the ground truth image.

The **multi-view consistency** of NeRFs, even for new poses, stems from the observation that the density of any point is independent of the viewing angle. Assume a point $x$ is visible from $n$ different cameras, then during training, the predicted density of that point converges to a value that is coherent with those $n$ observations. When viewed from a new perspective, we still obtain that exact same density value for $x$. However, we note here that a density that is coherent with the observations in the training data does not have to correspond to the real density at that point. This may be the case for example when the training data only covers limited angles.

### 3.3.1 General Architecture

Formally, a NeRF is a differentiable function $\theta$ of the form

$$\begin{aligned} \theta\colon \mathbb{R}^{d_x} \times \mathbb{R}^{d_d} &\longrightarrow \mathbb{R}^3 \times \mathbb{R} \\ \mathbf{x}, \mathbf{d} &\longmapsto c, \sigma \end{aligned}$$

where $\mathbf{x} \in \mathbb{R}^{d_x}$ and $\mathbf{d} \in \mathbb{R}^{d_d}$ are encodings of the coordinates $x \in \mathbb{R}^3$ in space

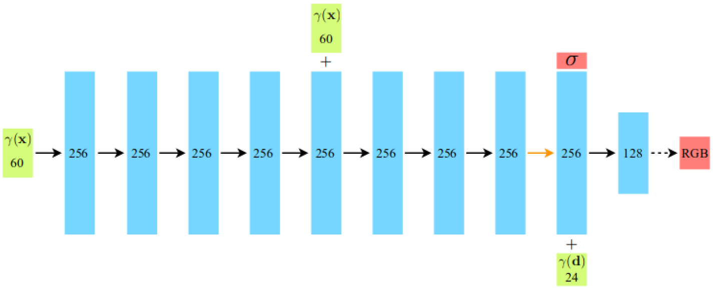

Figure 3.3: The original NeRF architecture consists solely of fully-connected ReLU layers. Only the color values are conditioned on the viewing direction. Credits: [12]

and the direction of the current ray $d \in \mathbb{R}^3$ (encodings for positional, directional and temporal features are explained in more detail in section 3.3.4). In practice, $\theta$ is implemented as a multi-layer perceptron, and the color is conditioned on the viewing direction to capture view-dependent effects such as specularities. In the original implementation, eight fully-connected ReLU layers are used to predict the density, followed by two additional layers to predict the RGB color values, as seen in figure 3.3.

### 3.3.2 Rendering an Image

The ray $\mathbf{r}$ at pixel $(u, v)$ is given by the camera intrinsic $K$ and pose $E$ through inverse projection, and is bounded by the near and far plane camera parameters $t_n, t_f$. The RGB radiance for this ray is then given by

$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))c(\mathbf{r}(t), \mathbf{d})dt$$

where $T(t) = \exp(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))ds)$ is the accumulated transmittance [12]. This integral is numerical approximated through $N$ finite samples along the ray. $\sigma_i$ and $\mathbf{c}_i$ are the density and color values of sample $i$, and the density is weighted based on how far the next sample is away, i.e. $\delta_i = t_{i+1} - t_i$ [12]:

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i(1 - \exp(-\sigma_i \delta_i))\mathbf{c}_i, \quad T_i = \exp(-\sum_{j=1}^{i-1} \sigma_j \delta_j).$$

### 3.3.3 Temporal and Expression Dependencies

So far, we have assumed that the scene is static, that is, that the visible objects in the scene do not move or deform with respect to the global frame of reference. This assumption has allowed us to restrict the problem of reconstruction to four degrees of freedom. However, the previous architecture fails once objects deform between captures, as there no longer exists a multi-view coherence. This can be the case in certain videos, or, as targeted by this thesis, for facial avatars.

As seen in the literature, a significant area of research focuses on temporal dependencies, as the applications are more general. However, some of the ideas can be trivially extended to conditioning the network on other factors, such as facial expressions.

In the basic setting, as introduced by [21], the standard NeRF is extended by an additional deformation module $\theta_d$

$$\begin{aligned} \theta_d : \mathbb{R}^{d_x} \times \mathbb{R}^{d_t} &\longrightarrow \mathbb{R}^3 \\ \mathbf{x}, \mathbf{t} &\longmapsto \Delta x \end{aligned}$$

which predicts a translation vector $\Delta x$ given a coordinate and time encoding. The purpose of this network is to predict how it needs to be deformed so that it maps to a canonical space for a given point, and therefore acts as a correspondence map. This approach is semi-supervised, as we have no ground-truth correspondence information, only the training images. To ensure that predicted deformations are realistic they can be regularized to ensure the deformation field primarily consists of rotations and translations, and are note excessive in magnitude, for example by using a divergence and offset loss [24].

### 3.3.4 Positional, Directional and Temporal Encodings

Spectral bias [38] is an effect observed with deep networks where the model is biased towards learning low-frequency functions in the data. Assuming a target function $\lambda : \mathbb{R}^d \to \mathbb{R}$, an encoding $\gamma : \mathbb{R}^d \to \mathbb{R}^m$ and a solution $f : \mathbb{R}^m \to \mathbb{R}$, $\lambda$ can be better generalized through $f$ if the complexity of the data manifold is increased through $\gamma$, i.e. such that $\lambda = f \circ \gamma$.

The scenes that we attempt to reconstruct often contain many high-frequency details - from fine textures to detailed geometry, especially for realistic facial avatars. At the same time, the problem domain is low-dimensional, varying only in the three positional coordinates, and potentially viewing direction and time.

Neural networks - the basis of NeRFs - are unable to capture these high-frequency details [14]. The researchers show that this is not just the case for neural radiance fields, but other types of neural representations as well such as images, relating to the earlier observation on spectral bias [38]. To address this issue, they introduce a Fourier feature mapping to artificially elevate the low dimensional input domain to a higher dimensional space:

$$\begin{aligned}\gamma : \mathbb{R}^3 &\longrightarrow \mathbb{R}^{3\times L}\\ x &\longmapsto (\sin(2^0\pi x), \cos(2^0\pi x), \cdots, \sin(2^{L-1}\pi x), \cos(2^{L-1}\pi x))\end{aligned}$$

where $L$ is a tunable hyperparameter denoting the number of frequencies used. In the original NeRF publication [12], this encoding is used for both the encoding of position and viewing direction, and several non-rigid adaptations use Fourier encodings for time embeddings as well, such as [21, 25].

**Other types of encodings.**

One-blob encodings [39] are a generalization of one-hot encodings with $k$ bins, designed to extend their use to continuous variables as well. A Gaussian distribution is applied with mean $s$ (the variable we are encoding) and discretized across $k$ bins. This way, multiple adjacent entries can be activated by the neural network. Some works use a one-blob encoding to increase the dimensionality of the time parameter in non-rigid settings [26].

Parametric encodings have been used in more recent works, using additional data structures to store learnable parameters in grid or tree structures [16]. Unfortunately, such approaches have a large memory footprint of complexity $O(n^3)$ with respect to the scene size.

Multi-resolution hash encoding [16] builds on top of parametric encodings, but replaces memory-intensive grids with a hash table. Using a spatial hash function [16]

$$h(\mathbf{x}) = \left(\oplus_{i=1}^{d} x_i \pi_i\right) \mod T$$

where $\pi_{1..d}$ are large unique prime numbers, they map grid coordinates to a table with up to $T$ indices. For a point $x \in \mathbb{R}^3$, the hash encodings of the closest grid points are interpolated, weighted by their distance to $x$.

### 3.3.5 Performance Optimizations

A simple method to improve the speed of the model during both inference and training is to use more efficient ray sampling. To generate an image, as described in section 3.3.2, points are sampled along a ray - however, most of these points are located in empty space and therefore do not contribute to the final output.

In [16], ray marching is accelerated by constructing a discrete occupancy grid around the (rigid) scene. A single bit denotes whether a given grid cell is occupied in the scene or not. This occupancy grid is used and updated during training. While this method significantly reduces training times (from hours to seconds [16]), it is designed for static scenes.

[26] addresses this limitation for non-rigid scenes by taking the maximum occupancy of a grid cell over the entire time frame. While this works well for small deformations, the method is less effective when motions span the entire scene, as most of the scene ends up marked as occupied.

# 4 | Contributions and Results

In this chapter, we first motivate the the focus on efficient point sampling along the rays by comparing the performance of different models. We use this information in the following section to improve the efficiency of standard architectures by using known priors. The final chapter is chronologically intertwined with the development of the training and inference pipeline, but is dedicated to addressing errors in camera pose estimation.

## 4.1 Understanding Performance Bottlenecks in Non-Rigid Settings

As discussed in the literature review in chapter 2, very few prior works on non-rigid NeRFs focus on performance improvements, resulting in significantly slower inference times than their rigid counterparts - while Instant-NGP achieves rates of 60 FPS in static settings [16], non-rigid networks performed relatively poorly; that is, up until MoNeRF [26] claimed to achieve "real-time" performance for novel view synthesis on fixed-length videos in a Preprint in December 2022[1]. This led to the original hypothesis that the deformation network - which is evaluated separately for every sampled point, is likely the main driver for the differences in training and inference speed.

**Method.** While there are surveys which compare order-of-magnitude performance for the overall pipeline [40], to the best of our knowledge, there is no research so far which analyzes the component-wise performance of neural radiance fields. However, since the downstream task of video conferencing

[1] The code was not published until July 2023.

has a hard real-time requirement with low latencies, we investigate the components more closely.

We benchmark the forward step of three different architectures, Instant-NGP [16] for rigid scenes as a baseline, TiNeuVox [25] for non-rigid scenes and MoNeRF [26] for optimized non-rigid scenes.

**Implementation Details.** All models are tested on the Lego dataset, on either the static [12] or the non-rigid version [21]. For Instant-NGP, we benchmark the PyTorch implementation [41] rather than the original C++/CUDA implementation. As some of the code structures are rather different, we opt for adding timers to the source code to benchmark the desired categories rather than using a tracer. Some components may be run multiple times for each image: here, we sum across each call for a single image, and compute the mean and standard deviation across multiple images. The modified repositories for benchmarking are included in the code base.

**Results.** We observe that while the deformation network does affect the performance of the system, its impact is lower than anticipated. The rendering stage (not considering MLP inference) in turn is significantly more important than expected; TiNeuVox is more than two orders of magnitude slower than the other two methods, not considering model queries.

In the case of MoNeRF, the deformation network only adds $\tilde{4}$ ms to the inference time.

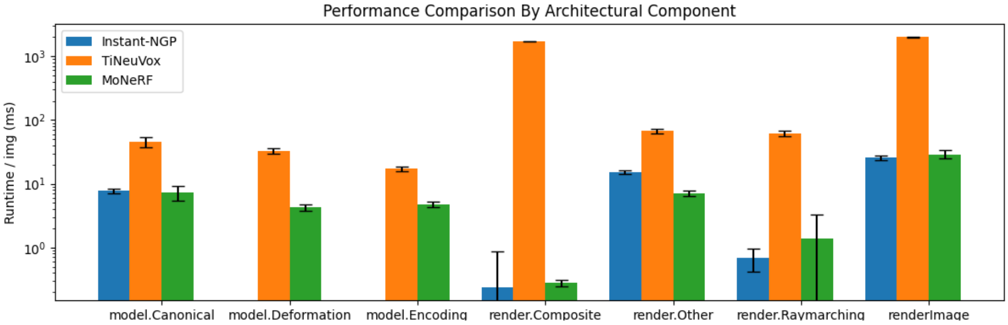

Figure 4.1: Average inference runtimes of different components in Instant-NGP, TiNeuVox and MoNeRF (log scale). *render.Raymarching*: sampling the points along the ray (e.g. knowing the occupancy, or just sampling uniformly). *render.Composite*: compute the pixel color values from the raw RGB and density information. *render.Other*: Python code, e.g. reshaping tensors etc.

| **identifier** | **Instant-NGP** | **TiNeuVox** | **MoNeRF** |
|---|---:|---:|---:|
| **model.Canonical** | 7.711 | 45.267 | 7.24 |
| **model.Deformation** | N/A | 32.921 | 4.255 |
| **model.Encoding** | N/A | 17.224 | 4.716 |
| **render.Composite** | 0.235 | 1718.304 | 0.276 |
| **render.Other** | 15.022 | 67.137 | 7.079 |
| **render.Raymarching** | 0.687 | 62.324 | 1.367 |
| **renderImage (total)** | 25.705 | 1987.959 | 28.962 |

Table 4.1: Runtimes of the different architectural components.

## 4.2 Performance Improvements Through Known Priors

While prior works have already dealt with facial avatars and novel pose and expression synthesis, they do not focus on performance. And while [26] achieves significant performance improvements in non-rigid settings by adapting ray sampling techniques from [16], their method is restricted to fixed-length videos.

Keeping track of a density grid allows more efficient ray sampling, which in turn results in lower inference times and more efficient compositing of the samples, as discussed in 3.3.5. The optimizations in [26], i.e. taking the maximum occupancy over time, would not work in our scenario: since we have no prior information on where the head will be located in the scene, we would need to set the occupancy to 1 everywhere. However, then we loose all benefits of using one in the first place.

There are three things we can take advantage of. We know that the subject in the scene is a human, we need to estimate the facial expression (to synthesize novel views for the currently observed frame) and we want to remove the background, since this would be undesirable in a virtual round table. Due to the first two, we need to estimate the current head pose and facial expression in any case, allowing us to reuse those computations for more efficient ray sampling since we know where the person is located. Finally, removing the background improves rendering times, as no ray sampling is required in that area.

### 4.2.1 Method

In the preprocessing stage, a user captures a short video clip of themselves. In this clip, the user rotates their head while speaking a few phrases. The images are fed through a pre-trained ResNet-50 architecture [36] to obtain the parameters of a 3DMM from which we extract the head pose and facial expression parameters - the choice of this model results from observations discussed in section 4.3. We remove the background using a pre-trained image-segmentation model from MediaPipe [42].

While the expression parameters are stored as part of the dataset, we use

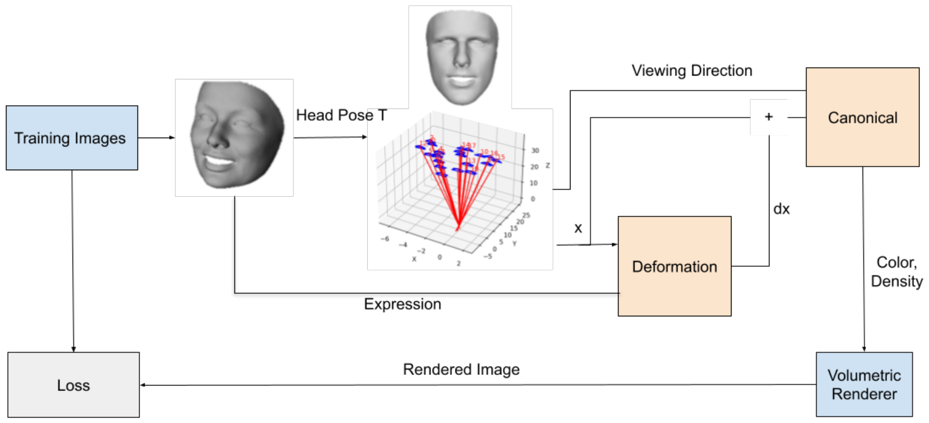

Figure 4.2: Overview of the training pipeline.

the head pose estimation $T \in \mathbb{R}^{4\times4}$ to fix and center the head. Using the head as a frame of reference, we convert the head pose parameters to camera extrinsic parameters (identical to the inverse, up to a correction term), which are stored as part of the dataset. The method is illustrated in figure 4.2.

The dataset is captured by a standard webcam; we then scale the images down to $224 \times 224$, as [36] is only trained on that image size. During training, we use a density grid identical to [16, 26].

### 4.2.2 Results

We illustrate the computed virtual camera poses from the normalized head pose in figure 4.3. The rays converge at the location of the normalized head position, as the head is centered in this stage of the pipeline.

Due to the change of reference, our occupancy grid is sparse and does not cover the entire motion path of the head, as illustrated in figure 4.4.

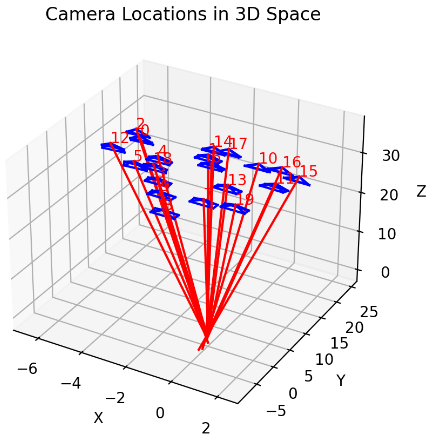

Figure 4.3: Moving from a fixed-webcam to a head-frame of reference: The blue rectangles are training images of the head in different poses, the red lines indicate their viewing direction of the aligned virtual cameras.

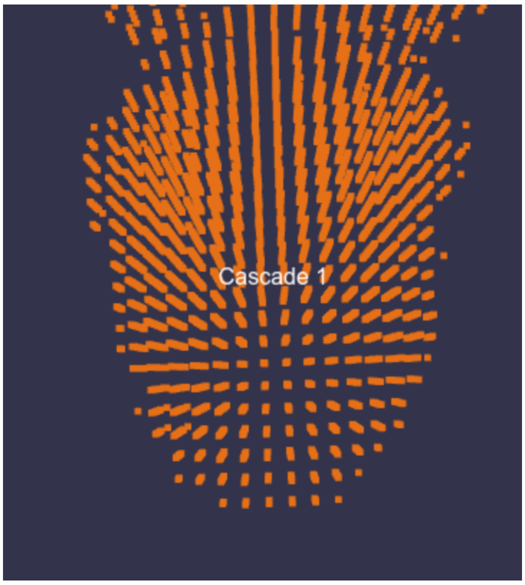

Figure 4.4: Occupancy grid of the head. Despite varying poses, the occupancy grid is sparse.

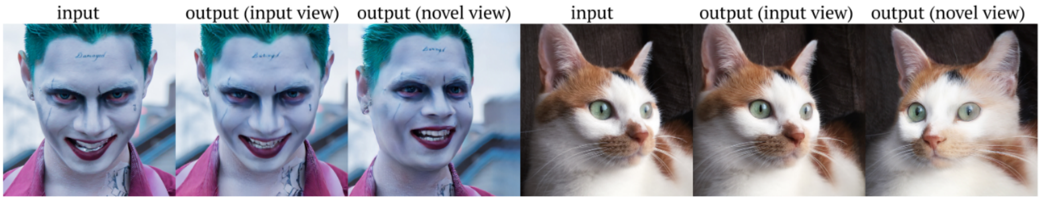

Figure 4.5: Results from NVIDIA, August 2023. The method is not restricted to human faces and only requires a single input view. Image from original publication [31]

Inference with our model runs at around 19-20 FPS on an RTX 3090 GPU, which is a bit slower than MoNeRF. This could be related to the processing of the expression features which are larger than the dimension of the time encoding in MoNeRF. Furthermore, the scene parameters are slightly different between the synthetic MoNeRF scenes and our captured images. Reparametrization might lead to small improvements.

**Comparison with NVIDIA.**

Our method requires the training of a NeRF for each person individually. This also involves capturing a short video of the person speaking some phrases while rotating their head. The recent work by NVIDIA [31], however, eliminates the need for a training stage, simplifying the usage. They do not rely on known priors, and can therefore adapt to different categories. Similarly, they are not restricted to the pure head region, as seen in figure 4.5. Finally, they achieve better performance (24 FPS) on the same hardware.

They rely on a pre-trained 3D-aware Generative Adversarial Network to train an efficient encoder architecture. The image is rendered at 128 × 128 pixels instead of 224 × 224, and then uses a super-resolution network to upscale the image to 512 × 512.

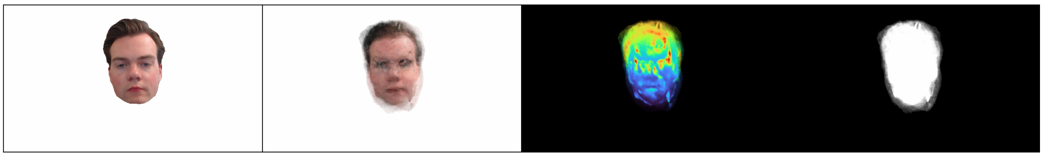

Figure 4.6: Using light-weight models for head-pose estimation is insufficient despite using additional alignment algorithms, resulting in blurred outputs. From left to right: ground truth, prediction, depth estimation, mask

## 4.3 Error Correction in Pose Estimation

While neural radiance fields rely on a multi-layer perceptron, that does not mean that they are good at handling even small noise in the data. Due to the hybrid nature of the approach, the system is very susceptible to camera calibration errors. This is linked with the observation that the same physical point will have two locations in the virtual space, as the rays of the cameras do not intersect at the correct point. The network, learning a density value for a given virtual point that corresponds to multiple physical points over time due to misalignment, returns blurred density and color values.

Initially, we used MediaPipe [42] to estimate the face mesh for training. This mesh differs from the BFM face model in the sense that it does not rely on PCA of identity and expression, and therefore does not clearly separate between the two. Combined with the fact that the light-weight model does not always predict the geometry accurately, the result is a face mesh whose geometry changes between each image.

The differences in geometry of the meshes lead to greater error terms when aligning the point clouds. The resulting estimated camera poses are noisy as well, leading to blurred reconstructions as seen in figure 4.6.

We therefore adopted two different pose estimation methods, one for training

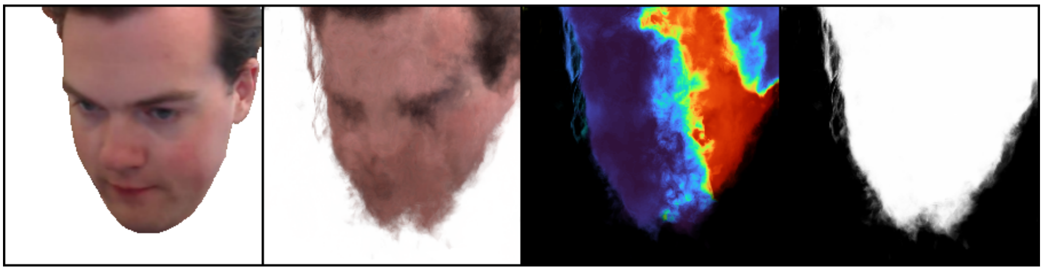

Figure 4.7: Using a 3DMM for head pose estimation is consistent in geometry, but initially not in alignment. From left to right: ground truth, prediction, depth estimation, mask

and another for inference. The latter continues to use MediaPipe for performance reasons, but during training where low latency is not as important we use the BFM 3DMM with parameter estimation through a Resnet-50 model [36] as already foreshadowed in section 4.2.

However, the estimated head poses are still not entirely accurate. Let $T_t \in \mathbb{R}^{4\times4}$ be the estimated head transformation matrix for time $t$, and $X_t \in \mathbb{R}^{4\times n}$ be the set of $n$ (static, e.g. forehead and ears) points of the face at time $t$ in homogeneous coordinates. If we had a perfect reconstruction, we should have

$$X_{canonical} = T_0^{-1}X_0 = \cdots = T_t^{-1}X_t$$

However, this is not the case - again, due to slight offsets and leading to blurred results as seen in figure 4.7.

We therefore compute an error correction term $E_t$ for all time steps $t$ such that

$$X_{canonical} = T_0^{-1}X_0 = E_1T_1^{-1}X_1 = \cdots = E_tT_t^{-1}X_t$$

Since $T_0^{-1}X_0$ is known, computing $E_1, ..., E_t$ can be computed using singular value decomposition. The resulting mesh-aligned camera extrinsic matrix - $E_tT_t^{-1}E_{canonical}$, where $E_{canonical}$ are the camera extrinsics at time 0 - leads

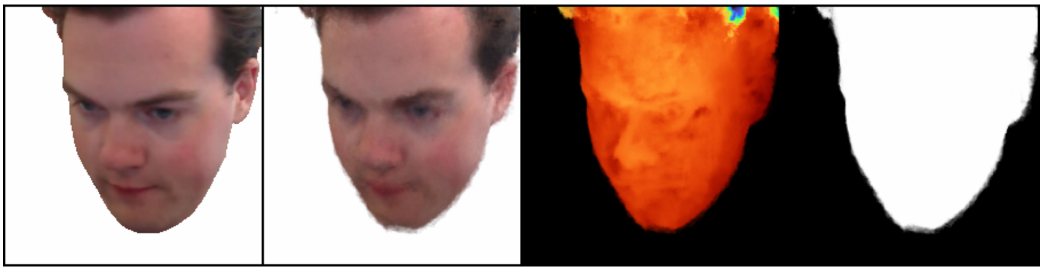

Figure 4.8: Manual alignment procedure applied to 3DMM head pose estimation result. Validation data is sharper, and the depth estimation is more consistent. From left to right: ground truth, prediction, depth estimation, mask

to sharper results when rendering the neural avatar from new perspectives or expressions, as seen in figure 4.8.

# 5 | Future Work

The opportunities of the technology open up a variety of possible extensions of this work. Firstly, as this project is application-focused, there are several interesting questions related to neural rendering that should be addressed when moving to a production setting; specifically concerning lighting and the removal of VR headsets. We elaborate on these questions in more detail.

Secondly, we discuss some ideas for generalized performance improvements that are not restricted to priors. VR video conferencing can benefit from these as well, effectively extending the capabilities of the system beyond faces - from upper body motion to bringing in objects that are shown in a call.

## 5.1 Topological Neural Radiance Fields

Neural Radiance Fields, being based on neural networks, are well suited for modeling continuous functions. High-frequency positional encoding was introduced in [14] to ensure that abrupt changes in a small neighborhood, as it often is the case with textures, could be reasonably well approximated by continuous functions.

In non-rigid settings, the requirements for modeling large changes in small neighborhoods may be less apparent. After all, most deformations are continuous. If you lift your arm, you do so continuously, if a person walks along a path, that too is a continuous deformation.

So when, if at all, do we encounter discontinuities? This is the case with very abrupt changes in the structure of the scene. Abrupt changes in this context can refer to breaking objects, or, as relevant to us, the opening of a persons mouth. Current methods fail here due to the following intuitive observations, considering a scene where an object is being broken apart:

- There is a point at which the deformation (from the canonical object) is zero - points left of the point are being torn to the left, and points right to it to the right
- The density of the points to the left and to the right are well defined, and correspond to the density of the canonical mesh
- However, the density at the tearing points is not well defined.

If defining a topological space for every time step in a non-rigid scene, it would *intuitively* make sense that NeRFs would only be able to model non-rigid scenes where the set of these spaces is a topological equivalence class, that is, there exists a homeomorphism between them. This is because NeRFs are based on MLPs, which in turn means that they are naturally continuous. It would be interesting to formalize the topology of non-rigid scenes, and prove that current architectures are not fully expressive in cases of non-homeomorphic changes in topology, as in the example described above.

In 2023, [43] proposed a novel topology of Neural Network layers, allowing them to approximate and detect discontinuities. In a second stage, we would be keen to investigate the impact such a framework could have on the expressiveness of NeRFs, as well as implement and test such architectures.

## 5.2 Remaining Problems for Production Usage

**Relighting.** The current architecture does not disentangle lighting from the scene, that is, it renders the person with the lighting that was present during

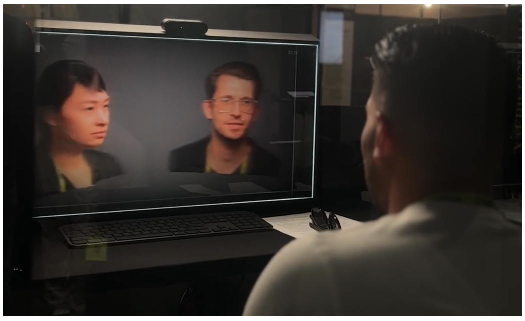

Figure 5.1: NVIDA presents a proof of concept of AI mediated 3D Video Conferencing at SIGGRAPH 2023, using results from [31]. Standardized lighting and a virtual room which does not embed the avatars into a scene reduces needs to relight the avatar. Image credits: NVIDIA

the training stage. This may not be too problematic in controlled setups similar to the one of NVIDIA as seen in figure 5.1 - due to the standardized lighting setup and the fact that the avatars just float in space reduces the sense of unrealistic lighting conditions.

However, in practice, lighting quality can differ tremendously between participants in a call. Furthermore, as soon as the neural avatars are embedded into a virtual office space, the neural avatars will need to be relighted to match the scene lighting.

Prior work in the field of lighting disentanglement has been done before as discussed in section **??**, however, more care is required to ensure accurate color calibration between the participants as a collective. Minor color discrepancies due to over-exposures or other factors become more noticeable when compared side-by-side - as it would be the case here - than in single-scene relighting tasks.

**Issues when wearing a VR headset.** In an actual virtual reality video conferencing system, unless using a light field display, users will be wearing a VR headset. These headsets cover significant portions of the face during the call, and need to be removed. Since during the inference stage, our system only relies on facial expression detection, we do not need to worry about removing the headset from the webcam image, as the NeRF deals with this naturally. However, accurately capturing a persons' expression including eye motion is necessary, and may not always be possible unless the VR headset is equipped with the necessary sensors itself.

Of course, further reducing the latency between webcam capture and rendered neural avatar will always remain essential. Some (more general) ideas on improving the performance for deformation predictions are discussed in section 5.3.

## 5.3 Generalized Performance Improvements for Deformations

While the performance improvements introduced in this thesis are application-specific, we came across some ideas to improve the performance of the deformation network in more general settings.

### 5.3.1 Sparse Grid Deformations and Dual Quaternion Blending

In all works that we have come across, the deformation is computed for each sampled point on the rays individually. However, if we assume that neighboring points deform similarly, one might be able to leverage a sparser representation of deformations.

Taking inspiration from the earlier classical work on non-rigid 3D reconstruction, DynamicFusion [44], we propose to further investigate the usage of dual-quaternion blending for a set of dynamically changing set of deformation nodes. Both their non-linear blending capabilities and computational efficiency could reduce the overhead of the current deformation networks. In DynamicFusion, deformation nodes are added based on where they are needed, and the warp function for arbitrary points $x_c$ is computed from the neighboring deformation nodes:

$$\mathbf{DQB}(x_c) = \frac{\sum_{k \in N(x_c)} \mathbf{w}_k(x_c)\hat{\mathbf{q}}_{kc}}{|| \sum_{k \in N(x_c)} \mathbf{w}_k(x_c)\hat{\mathbf{q}}_{kc}||}$$

where $N(x)$ returns the $k$ nearest transformation nodes of $x_c$. The result can be converted back to an $SO(3)$ transformation matrix.

In the original DynamicFusion implementation, the idea of using a grid was rejected due to memory efficiency issues [44]. This was due to the fact that all transformations had to be stored in memory, unfeasible even for low resolution grids - $6 \times 256^3$ parameters per frame for a voxel grid resolution of 256.

However, using a learning-based approach, we could train a deformation network to predict the transformations conditioned on time - but only for relevant sparse grid coordinates instead of for every sample. The exact transformation for the individual samples could then be computed using dual quaternion blending rather than inference on the deformation network.

### 5.3.2 Deformation Hash-Encoding

In practical settings, many deformations are similar: if a person lifts their arm, then all points along the arm could be transformed using the same transformation matrix. It may be worth investigating learning hash functions that map points with the same deformation to the same index.

# 6 | Conclusion

## 6.1 Summary of Results

We have demonstrated the potential of neural radiance fields for the usage in virtual reality video conferencing, both from a procedural perspective and also by identifying and addressing bottlenecks. We observe that while the deformation network does affect the performance of the system, the overhead is negligible compared to that of an un-optimized rendering function. We use this observation to create an optimal density grid given known priors - human avatars, with which we can leverage efficient volume rendering techniques. Our method can render novel views at approximately 20 FPS on consumer GPUs.

## 6.2 Limitations

While our method performs well under extreme head poses due to our different choice of coordinate system, and achieves competitive speeds during inference, our system comes with some limitations. Firstly, we have restricted ourselves to novel-view and pose synthesis of the head from the start, not considering the upper body or neck region. Just like other related works that use NeRFs [27, 30], we also assume that hair is tied up or rigid with respect to the head. Estimating the hair flow is a challenging task, especially in real-time settings where a simulation engine might be required.

We must furthermore acknowledge the fact that our work is no longer state of the art - the very recent work by NVIDIA [31] outperforms our results both in terms of visual quality and performance. As their work was published on

August 6, 2023, pivoting the thesis was no longer possible.

## 6.3 Ethical Considerations

**Data Handling.** Due to the nature of neural radiance fields, they do not have to trained on larger datasets. Instead, they are retrained from scratch for each scene individually. The person who we have captured the videos with has consented to the usage of the images for this thesis, however, the dataset will not be released to the public[1].

**Societal Impact.** The research was conducted with the intent to improve the quality of communication between people and further reduce the need for travel. However, as with many technologies, our results could be misused. We highlight potential issues so that they may be addressed by regulatory bodies or research fields at the intersection of law or ethics with technology.

Specifically, our technology could facilitate impersonation. Using publicly available images of a person, a neural avatar can be easily trained to then be used in video conferencing systems. Anyone could join a video call with their webcam and speak normally while the system renders the neural avatar - the fake identity - with the real expressions corresponding to the speech.

This opens up entirely new array of attack possibilities, changing the way we think about social engineering and fraud. Assuming continued improvement in the technology, one such attack might look as follows: an attacker schedules a meeting with a senior company employee, pretending to be a board member. The employee might not be as suspicious as with phishing emails, since this email is not requesting the person to download any files, or asking

[1] It is, however, included with the submission for reproducibility

for personal or confidential information. The call is taking place through a video calling platform trusted by the company, so again, no reason for the employee to become suspicious. Then the virtual meeting takes place - the (fake) board member joins the call, and after exchanging pleasantries, the attacker starts asking the employee about current status updates from the team for a supposedly upcoming board meeting. The employee - having at the very least seen the board member before - trusts the neural avatar of the board member and reveals confidential information, never questioning the authenticity of the person because they see a live video of them.

This attack may look very similar for phishing, making such attacks seem significantly more trustworthy. Whether it is a friend or family member being stuck abroad needing money - once we set up a call with them and see them "live", why would most people doubt that this is a fake?

We will require new ways of ensuring trust with these technologies. Unfortunately, since the neural rendering solution for facial avatars can be fed into any existing video conferencing as a virtual webcam, it is unlikely that platforms such as Zoom, Google Meet or Microsoft Teams will be able to detect such scams out of the box. One potential solution is inspired by a similar issue observed with deep fakes in journalism. Certain manufacturers have created cameras that cryptographically sign the photograph, allowing people to verify that the image was actually captured and not generated. Webcam manufacturers could similarly sign the outgoing video feed - signatures that common video conferencing platforms could verify to ensure the authenticity.

# Bibliography

1. McGillem, C. D. in *Encyclopedia Britannica* (May 17, 2023). https://www.britannica.com/technology/telegraph.
2. Borth, D. E. in *Encyclopedia Britannica* (July 7, 2023). https://www.britannica.com/technology/telephone.
3. Packetizer, I. *A history of video conferencing (VC) technology* https://www.packetizer.com/voip/history-of-videoconferencing/.
4. Borth, D. E. in *Encyclopedia Britannica* (July 28, 2023). https://www.britannica.com/technology/videophone.
5. Orts-Escolano, S., Rhemann, C., Fanello, S., Chang, W., Kowdle, A., Degtyarev, Y., Kim, D., Davidson, P. L., Khamis, S., Dou, M., Tankovich, V., Loop, C., Cai, Q., Chou, P. A., Mennicken, S., Valentin, J., Pradeep, V., Wang, S., Kang, S. B., Kohli, P., Lutchyn, Y., Keskin, C. & Izadi, S. *Holoportation: Virtual 3D Teleportation in Real-time* in *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (Association for Computing Machinery, New York, NY, USA, Oct. 16, 2016), 741–754. ISBN: 978-1-4503-4189-9. https://doi.org/10.1145/2984511.2984517 (2023).
6. *ArtStation - Ultimate AAA Character Creation Tutorial Course* ArtStation. https://www.artstation.com/marketplace/b/Wpz/ultimate-aaa-character-creation-tutorial-course (2023).
7. *Horizon Workrooms virtual office and meetings | Meta for Work* Meta. https://forwork.meta.com/horizon-workrooms/ (2023).
8. *MeetinVR - Business Meetings & Collaboration in VR* https://www.meetinvr.com/ (2023).

9. *Introducing Microsoft Mesh | Here can be anywhere.* https://www.microsoft.com/en-us/mesh (2023).

10. Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A. & Fitzgibbon, A. *KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera* in *Proceedings of the 24th annual ACM symposium on User interface software and technology* (Association for Computing Machinery, New York, NY, USA, Oct. 16, 2011), 559–568. ISBN: 978-1-4503-0716-1. https://doi.org/10.1145/2047196.2047270 (2023).

11. Choy, C. B., Xu, D., Gwak, J., Chen, K. & Savarese, S. *3D-R2N2: A Unified Approach for Single and Multi-view 3D Object Reconstruction* Apr. 1, 2016. arXiv: 1604.00449[cs]. http://arxiv.org/abs/1604.00449 (2023).

12. Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R. & Ng, R. *NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis* in *ECCV* (2020).

13. Garbin, S. J., Kowalski, M., Johnson, M., Shotton, J. & Valentin, J. *FastNeRF: High-Fidelity Neural Rendering at 200FPS* Apr. 15, 2021. arXiv: 2103.10380[cs]. http://arxiv.org/abs/2103.10380 (2023).

14. Tancik, M., Srinivasan, P. P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J. T. & Ng, R. *Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains* June 18, 2020. arXiv: 2006.10739[cs]. http://arxiv.org/abs/2006.10739 (2023).

15. Yu, A., Fridovich-Keil, S., Tancik, M., Chen, Q., Recht, B. & Kanazawa, A. *Plenoxels: Radiance Fields without Neural Networks* Dec. 9, 2021. arXiv: 2112.05131[cs]. http://arxiv.org/abs/2112.05131 (2022).

16. Müller, T., Evans, A., Schied, C. & Keller, A. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *ACM Transactions on Graphics* **41,** 1–15. ISSN: 0730-0301, 1557-7368. arXiv: 2201.05989[cs]. http://arxiv.org/abs/2201.05989 (2023) (July 2022).

17. Takikawa, T., Evans, A., Tremblay, J., Müller, T., McGuire, M., Jacobson, A. & Fidler, S. *Variable Bitrate Neural Fields* June 15, 2022. arXiv: 2206.07707[cs]. http://arxiv.org/abs/2206.07707 (2023).

18. Rivas-Manzaneque, F., Sierra-Acosta, J., Penate-Sanchez, A., Moreno-Noguer, F. & Ribeiro, A. *NeRFLight: Fast and Light Neural Radiance Fields Using a Shared Feature Grid* in. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2023), 12417–12427. https://openaccess.thecvf.com/content/CVPR2023/html/Rivas-Manzaneque_NeRFLight_Fast_and_Light_Neural_Radiance_Fields_Using_a_Shared_CVPR_2023_paper.html (2023).

19. Yariv, L., Hedman, P., Reiser, C., Verbin, D., Srinivasan, P. P., Szeliski, R., Barron, J. T. & Mildenhall, B. BakedSDF: Meshing Neural SDFs for Real-Time View Synthesis. Publisher: arXiv Version Number: 1. https://arxiv.org/abs/2302.14859 (2023) (2023).

20. Martin-Brualla, R., Radwan, N., Sajjadi, M. S. M., Barron, J. T., Dosovitskiy, A. & Duckworth, D. NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. *CoRR* **abs/2008.02268.** arXiv: 2008.02268. https://arxiv.org/abs/2008.02268 (2020).

21. Pumarola, A., Corona, E., Pons-Moll, G. & Moreno-Noguer, F. *D-NeRF: Neural Radiance Fields for Dynamic Scenes* Nov. 27, 2020. arXiv: 2011.13961[cs]. http://arxiv.org/abs/2011.13961 (2023).

22. Park, K., Sinha, U., Barron, J. T., Bouaziz, S., Goldman, D. B., Seitz, S. M. & Martin-Brualla, R. *Nerfies: Deformable Neural Radiance Fields*

Sept. 9, 2021. arXiv: 2011.12948[cs]. http://arxiv.org/abs/2011.12948 (2022).

23. Park, K., Sinha, U., Hedman, P., Barron, J. T., Bouaziz, S., Goldman, D. B., Martin-Brualla, R. & Seitz, S. M. *HyperNeRF: A Higher-Dimensional Representation for Topologically Varying Neural Radiance Fields* Sept. 10, 2021. arXiv: 2106.13228[cs]. http://arxiv.org/abs/2106.13228 (2023).
24. Tretschk, E., Tewari, A., Golyanik, V., Zollhöfer, M., Lassner, C. & Theobalt, C. *Non-Rigid Neural Radiance Fields: Reconstruction and Novel View Synthesis of a Dynamic Scene From Monocular Video* in *IEEE International Conference on Computer Vision (ICCV)* (IEEE, 2021).
25. Fang, J., Yi, T., Wang, X., Xie, L., Zhang, X., Liu, W., Nießner, M. & Tian, Q. *Fast Dynamic Radiance Fields with Time-Aware Neural Voxels* in *SIGGRAPH Asia 2022 Conference Papers* (Nov. 29, 2022), 1–9. arXiv: 2205.15285[cs]. http://arxiv.org/abs/2205.15285 (2023).
26. Kappel, M., Golyanik, V., Castillo, S., Theobalt, C. & Magnor, M. *Fast Non-Rigid Radiance Fields from Monocularized Data* Dec. 2, 2022. arXiv: 2212.01368[cs]. http://arxiv.org/abs/2212.01368 (2023).
27. Gafni, G., Thies, J., Zollhöfer, M. & Nießner, M. *Dynamic Neural Radiance Fields for Monocular 4D Facial Avatar Reconstruction* Dec. 5, 2020. arXiv: 2012.03065[cs]. http://arxiv.org/abs/2012.03065 (2023).
28. Weng, C.-Y., Curless, B., Srinivasan, P. P., Barron, J. T. & Kemelmacher-Shlizerman, I. *HumanNeRF: Free-viewpoint Rendering of Moving Peo-*

*ple from Monocular Video* June 14, 2022. arXiv: 2201.04127[cs]. http://arxiv.org/abs/2201.04127 (2023).

29. Jiang, W., Yi, K. M., Samei, G., Tuzel, O. & Ranjan, A. *NeuMan: Neural Human Radiance Field from a Single Video* Sept. 21, 2022. arXiv: 2203.12575[cs]. http://arxiv.org/abs/2203.12575 (2022).
30. Athar, S., Xu, Z., Sunkavalli, K., Shechtman, E. & Shu, Z. *RigNeRF: Fully Controllable Neural 3D Portraits* June 13, 2022. arXiv: 2206.06481[cs]. http://arxiv.org/abs/2206.06481 (2023).
31. Trevithick, A., Chan, M., Stengel, M., Chan, E. R., Liu, C., Yu, Z., Khamis, S., Chandraker, M., Ramamoorthi, R. & Nagano, K. *Real-Time Radiance Fields for Single-Image Portrait View Synthesis* in. SIGGRAPH (arXiv, May 3, 2023). arXiv: 2305.02310[cs]. http://arxiv.org/abs/2305.02310 (2023).
32. Hong, Y., Peng, B., Xiao, H., Liu, L. & Zhang, J. *HeadNeRF: A Real-time NeRF-based Parametric Head Model* Apr. 30, 2022. arXiv: 2112.05637[cs]. http://arxiv.org/abs/2112.05637 (2023).
33. Hughes, J. F. *Computer graphics principles and practice.* Third edition / John F. Hughes and six others. ISBN: 978-0-13-337370-7 (Addison-Wesley, Upper Saddle River, NJ, 2014).
34. cfr. *Answer to "More elegant way to achieve this same camera perspective projection model?"* TeX - LaTeX Stack Exchange. https://tex.stackexchange.com/a/323778 (2023).
35. Paysan, P., Knothe, R., Amberg, B., Romdhani, S. & Vetter, T. A 3D Face Model for Pose and Illumination Invariant Face Recognition. *2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance.* Conference Name: 2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)

ISBN: 9781424447558 Place: Genova, Italy Publisher: IEEE, 296–301. http://ieeexplore.ieee.org/document/5279762/ (2023) (Sept. 2009).

36. Deng, Y., Yang, J., Xu, S., Chen, D., Jia, Y. & Tong, X. *Accurate 3D Face Reconstruction with Weakly-Supervised Learning: From Single Image to Image Set* Apr. 9, 2020. arXiv: 1903.08527[cs]. http://arxiv.org/abs/1903.08527 (2023).

37. Tewari, A., Fried, O., Thies, J., Sitzmann, V., Lombardi, S., Sunkavalli, K., Martin-Brualla, R., Simon, T., Saragih, J., Nießner, M., Pandey, R., Fanello, S., Wetzstein, G., Zhu, J.-Y., Theobalt, C., Agrawala, M., Shechtman, E., Goldman, D. B. & Zollhöfer, M. State of the Art on Neural Rendering. *Computer Graphics Forum* **39.** _eprint: https://onlinelibrary.wiley.com/doi/p 701–727. ISSN: 1467-8659. http://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.14022 (2023) (2020).

38. Rahaman, N., Baratin, A., Arpit, D., Draxler, F., Lin, M., Hamprecht, F. A., Bengio, Y. & Courville, A. *On the Spectral Bias of Neural Networks* May 31, 2019. arXiv: 1806.08734[cs,stat]. http://arxiv.org/abs/1806.08734 (2023).

39. Müller, T., McWilliams, B., Rousselle, F., Gross, M. & Novák, J. *Neural Importance Sampling* Sept. 3, 2019. arXiv: 1808.03856[cs,stat]. http://arxiv.org/abs/1808.03856 (2023).

40. Gao, K., Gao, Y., He, H., Lu, D., Xu, L. & Li, J. *NeRF: Neural Radiance Field in 3D Vision, A Comprehensive Review* May 10, 2023. arXiv: 2210.00379[cs]. http://arxiv.org/abs/2210.00379 (2023).

41. *ngp_pl* original-date: 2022-07-27T08:06:44Z. Aug. 29, 2023. https://github.com/checkandvisit/3dml-instant-ngp-pytorch (2023).

42. Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., Zhang, F., Chang, C.-L., Yong, M. G., Lee, J., Chang, W.-T., Hua, W., Georg, M. & Grundmann, M. *MediaPipe: A Framework for Building Perception Pipelines* June 14, 2019. arXiv: 1906.08172[cs]. http://arxiv.org/abs/1906.08172 (2023).

43. Della Santa, F. & Pieraccini, S. Discontinuous neural networks and discontinuity learning. *Journal of Computational and Applied Mathematics* **419,** 114678. ISSN: 0377-0427. https://www.sciencedirect.com/science/article/pii/S0377042722003430 (2023) (Feb. 1, 2023).

44. Newcombe, R. A., Fox, D. & Seitz, S. M. DynamicFusion: Reconstruction and tracking of non-rigid scenes in real-time. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* Conference Name: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) ISBN: 9781467369640 Place: Boston, MA, USA Publisher: IEEE, 343–352. http://ieeexplore.ieee.org/document/7298631/ (2023) (June 2015).