Neural Radiance Field Driven Efficient and Controllable 3D Portrait Generation

by

Kuoyuan Li
Student number: 1072843

Supervised by Dr Mingming Gong,
Dr Kris Ehinger,
Dr Qiuhong Ke

A thesis submitted in total fulfillment for the
degree of Master of Computer Science

in the
Faculty of Engineering and Information Technology
School of Computing and Information Systems
THE UNIVERSITY OF MELBOURNE

October 2023

THE UNIVERSITY OF MELBOURNE

Abstract

Faculty of Engineering and Information Technology
School of Computing and Information Systems

Master of Computer Science

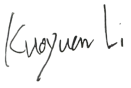
by Kuoyuan Li
Student number: 1072843

Generating 3D human face portraits has been a long-standing research topic in the field of computer vision, with application areas spanning mixed reality, live streaming, gaming, and online video chat. Users often expect to control and render portraits with arbitrary facial expressions and head poses in order to react accordingly to various scenarios, to express emotions, or to replicate the portrait movements of others. Recent advances in implicit neural rendering techniques, particularly the family of Neural Radiation Field (NeRF) based models, offer promising techniques for creating 3D scene images. However, for the rendering of human portraits using NeRF, despite the recent growing interest and several studies, there are still multiple issues that have not been explicitly addressed. In this study, we introduce a new controllable NeRF model tailored for human portraits, which can tackle some of the existing challenges and further advance the 3D portrait rendering technology. Our proposed model - 3DMM-guided Efficient Neural Radiation Field (3ENeRF for short) - features automatic rendering of dynamic 3D portraits and has a faster learning rate than its predecessors. The innovation of our approach hinges on three key advances: (1) an innovative data collection method that permits users to take portraits with basic devices such as mobile phones, thus obviating the need for dedicated filming settings and providing flexibility in different environments; (2) a pioneering 3D portrait image rendering model that facilitates the creation of photo-realistic images with adjustable facial expressions and poses in different contexts; and (3) the integration of an acceleration technique to increase training speed without compromising too much quality. Through our experiments to render human portraits under unseen viewpoints and exhibiting novel facial actions, our model has demonstrated its effectiveness and efficiency, marking a step forward in the field of 3D portrait rendering.

Declaration of Authorship

I, Kuoyuan Li, declare that this thesis titled, 'Neural Radiance Field Driven Efficient and Controllable 3D Portrait Generation' and the work presented in it are my own. I confirm that:

- this thesis does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any university; and that to the best of my knowledge and belief it does not contain any material previously published or written by another person where due reference is not made in the text.
- where necessary I have received clearance for this research from the University's Ethics Committee and have submitted all required data to the School.
- the thesis is 25,212 words in length (excluding text in figures, tables, code listings, bibliographies, and appendices).

Signed: 

Date: 16/10/2023

Acknowledgements

First, I would like to acknowledge and express my sincere gratitude to my research supervisors, Dr Mingming Gong and Dr Kris Ehinger. Dr Mingming Gong generously devoted his time, providing valuable advice, support, knowledge and insights as I conducted this study. He offered not only suggestions specific to this topic but also guidance on how to approach academic research and writing in general. His suggestions and advice are deeply appreciated. I would like to extend my gratitude to Dr Kris Ehinger. Although Dr Kris Ehinger had other much more important things in her life in the past one and a half years, she consistently made time for our supervision meetings and assisted in refining my thesis.

Due to external factors, I faced numerous challenges in the second half of my research, prompting me to switch to part-time study and extend my research by an additional semester. Both Dr Mingming Gong and Dr Kris Ehinger supported this decision and continued to supervise me until I completed this thesis. I am profoundly touched by their unwavering dedication and would like to extend my deepest gratitude to them.

I also want to express my great appreciation to my parents, who supported my tuition and living expenses in Australia. They stood by my decision to extend my studies and were a source of comfort during stressful times. I must also acknowledge my partner, whose continuous support and understanding were crucial during the challenging days and nights. I could not have completed this research and thesis without you.

Lastly, I want to thank all my friends and fellow students who provided immense support throughout the development of my thesis. Special mentions go to Tianyu Huang, who served as the second subject in our experiments, and Qianjun Ding, who was the subject to be reenacted in our experiments. I am also grateful to Ziyang Huang and Jiachen Li, who participated in my data collection, even though their contributions were not used in the final experiments. For all of my friends, your companionship has been invaluable and I could not have completed this study without any of you.

Contents

List of Tables

List of Figures

Abbreviations

3DMM	**3D M**orphable Face **M**odel
GAN	**G**enerative **A**dversarial **Network**
NeRF	**Ne**ural **R**adiance **F**ield
MLP	**M**ulti-**L**ayer **P**erceptron
CNN	**C**onvolutional **N**eural **N**etwork
SMPL	**S**kinned **M**ulti-**P**erson **L**inear

Chapter 1

Introduction

Faithful construction and manipulation of human portraits is a long and continuous research problem in the computer graphics and vision domains. A human portrait typically denotes a visual representation or depiction of an individual, often spotlighting the face while usually encompassing other characteristics such as the head, shoulders, neck, upper torso, and the surrounding background (figure 1.1). While portraits can be manifested in various forms, from paintings to sculptures, this research narrows its focus to photographic images of realistic individuals set within real-world contexts. Given a human portrait, the ability to enable full modification of its particular properties, such as 3D viewpoint, facial expression, and head pose, while maintaining photo-realistic fidelity (as shown in Figure 1.2), is critical in various domains like augmented reality

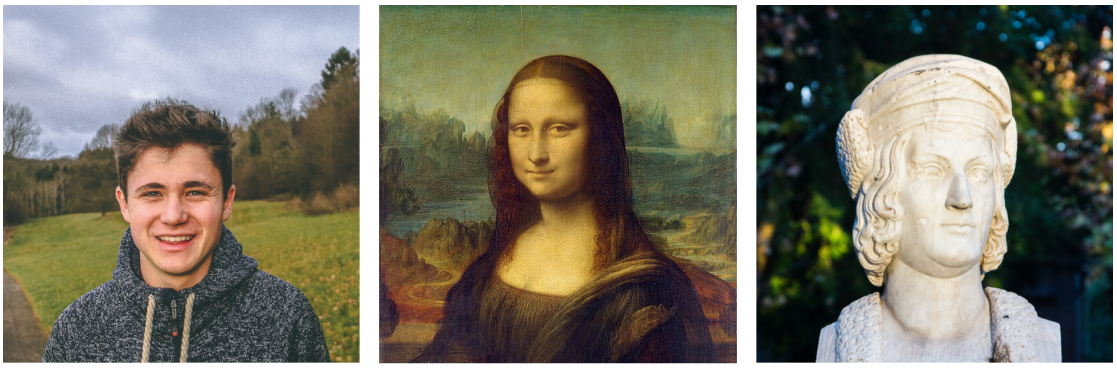


FIGURE 1.1: Portraits can be photos, paintings, or sculptures. Our research targets photographic human portraits only. Source: Fig. 1. fotosme21, https://pixabay.com/photos/nature-people-fall-a-grass-adult-3093158. Fig. 2. Da Vinci, L., https://commons.wikimedia.org/wiki/File:Mona_Lisa.jpg. Fig. 3. Blaz Erzetic, https://unsplash.com/photos/c-colombo-head-bust-vDPOywEkfbM.

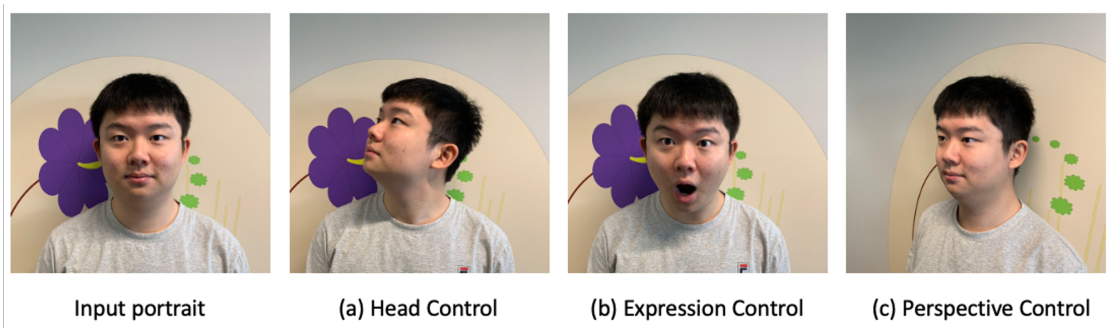


FIGURE 1.2: Given a human portrait, we want to synthesise new images with full control over the (a) head pose, (b) expression and (c) perspective, while maintaining realism.

(AR), virtual reality (VR), live streaming, face-time interactions, and the metaverse. For instance, within mixed reality, we may want to embed a human portrait seamlessly into an authentic environment. Such portraits should be adaptable per user preferences. As scenarios evolve, users might expect their facial expressions and head orientations to adjust accordingly or even emulate the actions of others. Such capabilities offer immense potential across entertainment, business interactions and numerous other applications. Crucially, these digitally crafted portraits should mirror reality, such that it is non-trivial to distinguish between synthetic or real scenes, ensuring a holistic 3D immersive experience.

Technologies such as Computer Graphics (CG) software allow for the manual creation of human portraits and are commonly leveraged in movies and games. Yet, these creations are handled by domain experts who are mastered in those specialized software applications, and it takes hours or even days for them to create a diverse set of portraits with varying appearances for a single identity. Manually editing each portrait every time a new expression or pose is desired is time-consuming and impractical. Meanwhile, despite the growing research interest and sustained efforts over the years, automatically generating novel human portraits remains a daunting challenge for several reasons: 1. Human portraits are set within real-world environments that are inherently 3D, indicating that 3D geometric constraints must be cautiously considered when creating portrait images from multiple views. 2. The appearance of a portrait is influenced not just by the facial 3D geometry but also by factors such as lighting conditions and skin reflectance under various viewpoints. It is essential to incorporate these factors when generating portrait images. 3. Portraits typically encompass elements beyond the face, like accessories, hair,

upper torso, and background. These components, also being 3D and intrinsically linked to facial area, must be generated synchronously to ensure a consistent representation.

In computer vision, the information required to render a portrait, such as the scene geometry and object texture, is commonly encapsulated in a computational model. These models, with their diverse architectures and underlying algorithms, allow users to query and manipulate them, thereby producing customized portraits tailored to their expectations. Over the past several decades, various classes of portrait models have emerged. One of the pioneering approaches is the family of 3D Morphable Face Models (3DMMs) [14]. 3DMMs are constructed using extensive 3D human facial scans, which are commonly represented as triangle meshes. These scans are paired with corresponding 2D photographs, capturing a wide range of identities, facial expressions, and poses. 3DMMs offer a parameterization of the human head, breaking it down into distinct characteristics such as face shape, facial expressions, and appearance, and facilitate their manipulation [14]. Generative Adversarial Networks (GANs), another prominent methodology for synthesizing novel images, have also been adapted for the controlled generation of unique human portraits [15, 16]. However, a limitation that arises with GANs is due to the 2D nature, they exhibit artifacts and suboptimal results when users attempt to control viewpoints. To address the challenges of constructing the 3D structure and appearance of scenes and facilitating the generation of novel viewpoints, recent research has introduced neural rendering models. Neural rendering integrates insights from physics-based computer graphics with the latest advancements in deep learning. They utilize neural networks to implicitly capture the detailed geometry and appearance of complex scenes, enabling the creation of photo-realistic images in a controlled manner [17]. A prime example of this innovation is the recently introduced Neural Radiance Field (NeRF) [4]. NeRF employs a single neural network to achieve an implicit volumetric representation of each entire static scene, bypassing the need for explicit 3D modelling using voxels, point clouds, or meshes. This method is not only more memory-efficient but also remarkably effective in synthesizing photo-realistic images and videos from previously unseen camera positions and view directions.

Over the past two years, multiple models have built upon the original NeRF to specialize in rendering human portraits, granting users the ability to manipulate specific properties [8, 9, 18, 19]. These controllable NeRFs for human portraits have demonstrated satisfying outcomes in terms of the quality of generated portrait images and the extent

of portrait manipulation, ranging from broader aspects like expression and pose to finer details such as blinking eyes or adjusting the mouth's shape. However, despite their advancements, current controllable NeRFs exhibit several limitations. Some necessitate additional annotations in the training data to facilitate manipulation, while others are limited to rendering portraits in isolation, either excluding the background entirely or only permitting integration with transparent backgrounds as an extra step. Certain models produce only low-resolution portraits, lacking in detailed facial features and rich backgrounds. Furthermore, a common challenge across most models is the extensive training duration, posing significant hurdles for real-world application deployment.

In this research, we introduce a novel controllable NeRF model tailored for human portraits. This model is designed for swift training and can render novel portrait images with satisfying quality. We argue the limitations of existing models stem from a critical but unresolved issue: the misalignment between the 3DMM space and the world space. This issue is further elaborated upon in section 4.2. We propose an innovative data collection and processing method to address this misalignment. Our proposed model can be efficiently trained using a monocular video capturing a specific individual from various angles. Once trained, users have the flexibility to adjust the attributes of the portrait, including expression, pose, and viewpoints. Importantly, our model achieves a faster training time compared to its contemporaries.

This thesis begins by outlining our research objectives. We then provide a literature review, covering existing methods for rendering human portraits and delving into NeRF models and their successors specifically related to human portraits. Subsequently, we elucidate our methodology, detailing both the model architecture and our innovative approach to data collection and processing. Our experimental results are presented, with comparisons drawn to selected baseline models. Finally, we conclude our research and suggest potential directions for future exploration.

Chapter 2

Research Objectives

Our research presents an innovative method that empowers users to create unseen portrait images or videos of any real person, granting them control over facial expression, head pose, and view direction. We outline our two main research objectives in detail below:

Objective 1: Build a NeRF-based model to allow users to control the generation of novel portrait images.

NeRF has proven to be an exceptional method for synthesizing photo-realistic images of static scenes from unseen viewpoints. It can implicitly capture 3D scene information, including geometry and appearance, with a memory-efficient approach. Encouraged by NeRF's remarkable rendering quality and ability to capture fine details, we aim to leverage it in this research. Our objective goes beyond merely generating novel viewpoints, we additionally seek to allow users to manipulate portrait expressions and poses. Human portraits present distinct challenges compared to static scenes. They involve dynamic elements, and the appearance may vary due to rigid transformations (e.g., head rotations) and non-rigid transformations (e.g., facial expressions). In our research, we define a portrait as the human head and a portion of the upper torso within a real-world environment, without imposing strict background limitations. Besides, when generating a portrait from a new viewpoint, the background viewpoint should also adapt correspondingly to ensure a seamless and photo-realistic transition. These requirements present a complicated and challenging task, and to the best of our knowledge, no existing

open-source approach has achieved this level of sophistication, inspiring our research topic.

Objective 2: Improve the training efficiency of our model.

An essential factor hindering NeRF's application in real-world scenarios is the extensive training time. Each scene in NeRF requires training from scratch, without any pretraining available. Consequently, for new scenes, on-the-fly training becomes necessary to provide an optimal user experience. To address this limitation, we explore a family of efficient NeRF variants (described in section 3.7) to improve training efficiency. However, none of the existing work has effectively incorporated the high efficiency of these models into the task of generating human portraits with background. Our research objective is to further enhance the efficiency of training a NeRF-based model from scratch for portrait images.

By tackling these objectives, our research seeks to push the boundaries of portrait generation, opening up new possibilities for creative facial variations and interactive control for users. We aspire to contribute novel insights to the field, bridging the gap between NeRF's capabilities and the dynamic world of human portraiture.

Chapter 3

Literature Review

Our research develops an innovative methodology that enables the rendering of human portraits with full control over the head pose, expression and viewpoint, as well as maintaining details with high efficiency during training. This research is highly related to preceding studies on 3D Morphable Face Models (3DMMs), Generative Adversarial Networks (GANs) and novel view synthesis via Neural Radiance Fields (NeRF).

3.1 3D Morphable Face Models

3D Morphable Face Models (3DMMs) are a family of facial models, with the initial model being introduced by Blanz and Vetter [1] (referred to as the "vanilla 3DMM" below). 3DMMs revolutionize the representation of human faces and play a vital role in a collection of human faces synthesis work [8, 20–23].

Vetter and Blanz devised a method to automatically derive the 3D model of a human face from its 2D image, where the 3D model supports the generation of realistic unseen human faces and visualization of existing human faces under unobserved viewpoints. The vanilla 3DMM splits the face into two components: shape and texture. The shape component, typically denoted by the coordinates of facial landmarks, encompasses details such as eye shape, mouth position and facial contour. The texture component entails colour information like the eye and skin colours mapped onto the facial surface. 3DMMs are statistical models built with face datasets made up of a wide range of 2D face images, along with corresponding head structure data derived from laser scans. The

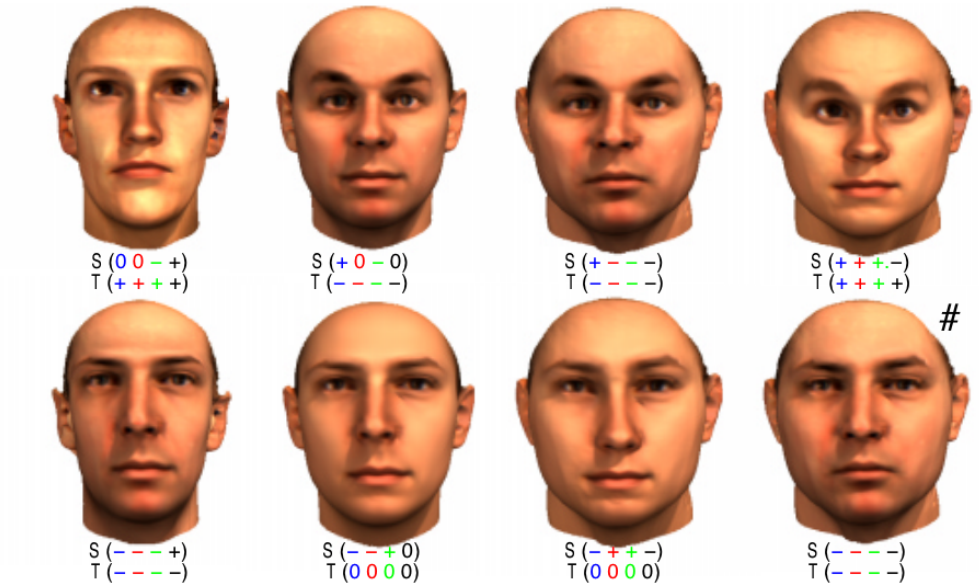


FIGURE 3.1: Varying the Shape (S) and Texture (T) parameters could result in different faces. Figure source: [1]

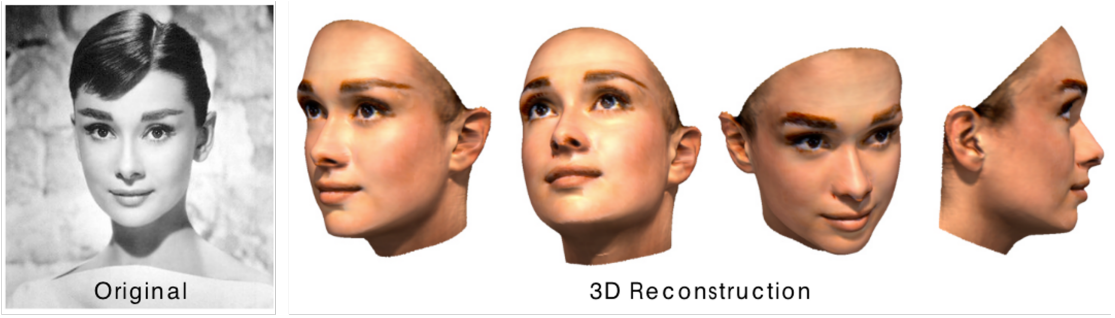


FIGURE 3.2: 3DMM can model 3D face from a 2D image and synthesis faces from novel viewpoints. Figure source: [1]

vanilla 3DMM generalizes the variations in the shape and appearance of human faces by encoding them into two sets of low-dimensional parameters. Each set of parameters is extracted by performing Principal Component Analysis (PCA) on the coefficients in a linear combination of facial shapes or textures from the dataset. By employing linear combinations of faces, the vanilla 3DMM enables us to modify the shape and texture coefficients (known as the parameters of the 3DMM) to generate 3D face models with fresh shapes and appearances, as illustrated in Figure 3.1. Additionally, it allows for the explicit modelling of the 3D mesh of facial images and rendering unseen viewpoints of faces, as shown in Figure 3.2. However, the vanilla 3DMM is trained with a relatively small dataset of 200 heads, limiting its capabilities of generalization. Moreover, it lacks control over the head pose and other details of observed human faces, which are essential for various applications.

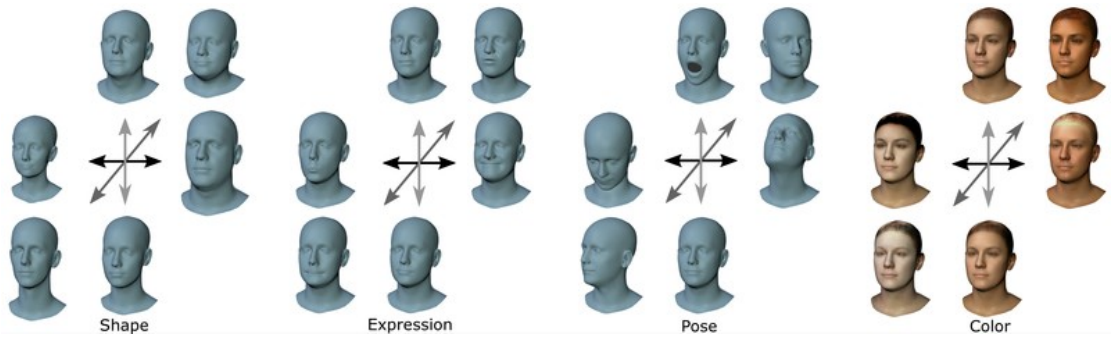


FIGURE 3.3: FLAME decomposes human head into shape, expression, pose and colour. Figure source: [2]

In recent years, with the advancements in neural networks, 3DMMs are studied in the context of deep learning [2, 14, 24–26]. FLAME [2] gathered three face datasets and implemented a model with over 33,000 3D face scans, thereby exhibiting improved generalizability to faces with diverse characteristics such as age, gender, and ethnicity. Furthermore, FLAME decomposes the representation of faces into four components: shape, pose (which emphasizes the movements of the jaw, neck, and eyeballs), facial expression and colour, as depicted in Figure 3.3. This decomposition enables richer control over facial properties when synthesized. Meanwhile, CoMA [26] observed that the linear combination of faces in the vanilla 3DMM fails to capture extreme facial deformations and exaggerated expressions. Hence it developed a framework based on a 3D convolutional neural network (CNN) encoder-decoder and trained with a facial dataset containing extreme expressions. The parameters derived from the encoder can be manipulated to generate faces with extreme expressions, wherein the non-linearity inherent in the neural network plays a crucial role. DECA [24] extends the capabilities of FLAME by addressing the challenge of capturing wrinkle changes associated with different facial expressions. DECA employs an encoder-decoder network that takes a 2D image as input and predicts a UV displacement which represents an expression-agnostic map that captures facial wrinkles. By combining the coarse geometry learned by FLAME with the UV map, DECA can synthesize faces with intricate details. MICA [25] tackles a notable limitation observed in prior models, which is the lack of metrical reconstruction in human facial modelling. When using a perspective camera, these models encounter ambiguity in determining the scale of the face and its distance from the camera solely based on the 3D scan. While achieving satisfactory 2D alignment, they often fall short in providing precise metrical 3D mesh representation and meaningful spatial placement.

To address this issue, MICA introduces a face dataset with distance supervision and trains a model using this dataset. The trained model generates meshes with accurate metrical information, ensuring improved spatial fidelity and meaningful positioning in 3D space.

In summary, 3DMMs could precisely parameterize human faces, allowing for the generation of 3D meshes from 2D images. They also provide the flexibility to manipulate specific properties like facial expression, texture, and pose. Nevertheless, these models encounter limitations when it comes to integrating additional components such as hair, mouth interior and background. Moreover, they are unable to capture variations such as light reflections due to alterations in viewpoints.

3.2 Generative Adversarial Networks

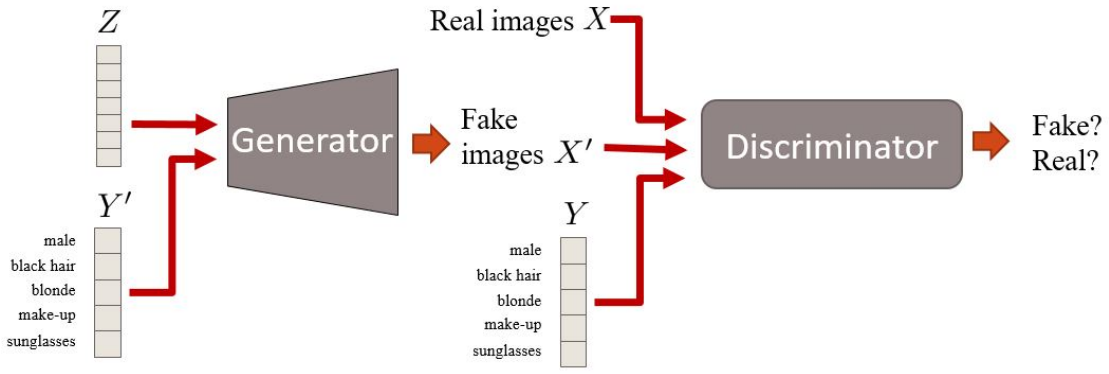


FIGURE 3.4: Architecture of Conditional Generative Adversarial Networks. Figure source: [3]

In addition to the 3DMMs, deep generative models have been vastly adapted to conduct human portrait synthesis. Numerous models are developed based on the Generative Adversarial Networks (GANs) [3, 27] and Conditional Generative Adversarial Networks (cGANs). GANs [27] leverage a dual-player framework, comprising a generator and a discriminator. The generator produces data instances, while the discriminator evaluates their authenticity. This adversarial interplay stimulates an iterative refinement process for both networks, thereby yielding increasingly realistic outputs. cGANs [3] take this concept further by incorporating additional information, such as class labels or attribute vectors, into both the generator and discriminator as illustrated in figure 3.4. This

enrichment empowers the model to proficiently generate data instances aligned with specified data categories or imbued with distinct features.

GANimation [16] introduces a GAN conditioning scheme based on Action Units (AU) annotations. AUs provide a representation of the facial muscle movements that constitute human expressions. Its GAN model is conditioned on a one-dimensional vector indicating the presence or absence of specific action units and the magnitude of their influence. This approach allows for the control of facial muscle movements through the manipulation of AUs with varying combinations, thereby enabling the generation of unobserved expressions. DefGAN [15] similarly leverages Action Units (AU) and proposes a two-phase architecture: A motion editing phase and a texture editing phase. The motion editing phase involves training a Convolutional Neural Network (CNN) to understand facial movements by predicting a deformation field that can transform the input face to match a target expression. This deformation field is predicted by conditioning the subject AUs and the target AUs. The Texture Editing Phase involves another CNN that operates on the deformed image and focuses on adding or editing textures, such as teeth or shadows, to complete the editing process. Both phases of the model are optimized with a discriminator network in an end-to-end manner. This adversarial training scheme helps the model generate more realistic and high-quality edited images by ensuring they are consistent with human facial expressions and textures.

In contrast, ConfigNet [28] takes a distinct approach without relying on Action Units (AU). It achieves fine-grained control over the generated faces, including various aspects such as head pose, hairstyle, facial hair style, expression, and illumination, by factorizing the latent space within the cGAN model into separate and clearly defined parts, where each corresponding to a specific aspect of face images. By doing so, ConfigNet can manipulate these factors independently, allowing for precise and customized control over the various attributes of the generated faces. DisentangledGAN [21] follows a similar disentanglement approach as ConfigNet. It decomposes a portrait into identity, expression, pose, and lighting attributes, allowing for the synthesis of new images conditioned on these individual factors. One notable contribution of DisentangledGAN is its utilization of a pretrained 3DMM as a prior. This prior guides the generator in producing images that mimic the characteristics and variations represented by the 3DMM. This incorporation of a prior model helps ensure that the generated images conform to

a realistic and structured representation of human faces, enhancing their quality and adherence to facial attributes.

However, these models do not align with our research objectives. They are fundamentally reliant on 2D image data for training and are constructed using models designed for 2D information, such as Convolutional Neural Network (CNN) models. They lack crucial 3D information about the scene, such as the geometry and appearance under varying viewpoints. This limitation poses challenges when it comes to controlling the 3D geometric properties of portraits and freely altering camera viewpoints. Additionally, these models are primarily focused on the human face alone, neglecting other aspects of the portrait like the upper torso and background information. Therefore, our research will bypass GAN-based and other 2D-centric approaches.

3.3 Neural Radiance Field

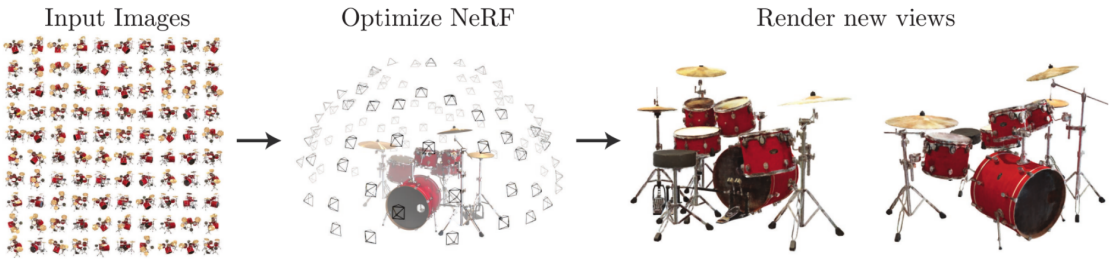


FIGURE 3.5: Optimized NeRF can render new views of the scene. Figure source: [4]

Neural radiance field (NeRF) [4] has achieved astonishing results on novel view generation for static scenes. Given a sequence of RGB images (such as video frames) of a single scene from different viewpoints and corresponding camera information, NeRF optimizes a multilayer perceptron (MLP) to learn the volumetric representation of the scene. With the learned volumetric representation, users can generate scenes from viewpoints that do not exist in the input images (figure 3.5). The camera information includes intrinsic and extrinsic parameters, which can be easily obtained for synthesized 3D objects created by software like Blender [1]. For videos filming real-world scenes, the camera information of each frame can be inferred using Structure-from-Motion (SfM) and Multi-View Stereo (MVS) models such as Colmap [29].

[1] Blender is a well-known software used for creating 3D objects, available at https://www.blender.org/

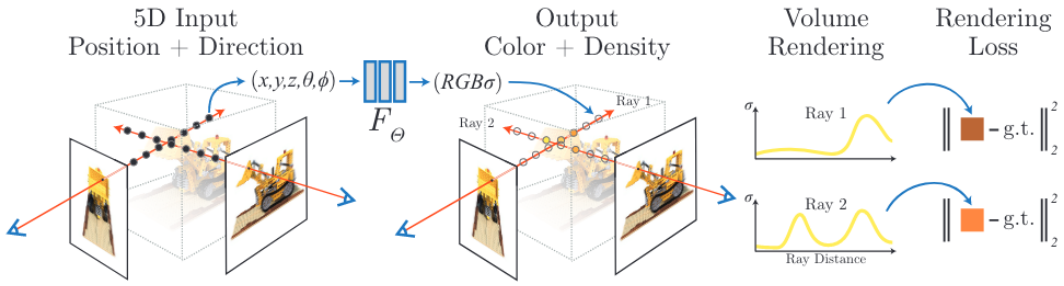


FIGURE 3.6: The architecture of the original NeRF model. Figure source: [4]

Figure 3.6 presents the architecture of the NeRF model. In NeRF, rays are traced based on the intrinsic and extrinsic parameters of each camera where each ray corresponds to a pixel. The colour of each ray/pixel is computed using the classical volume rendering, integrating continuous points along the ray with the formula 3.1. In the formula, C refers to the accumulated colour along a ray. t_n and t_f are the near and far bounds of the ray respectively. $r(t)$ represents the position of a 3D point along the ray at parameter t. $T(t)$ represents the transmittance along the ray up to a point with parameter t, describing how much light survives as it travels through the medium up to that point. $\sigma(r(t))$ is the density of a point $r(t)$, which describes the amount of radiance accumulated as the ray passes through it, where a density of zero indicates an empty point, while a density of one indicates a solid object that blocks further emission along the ray. $c(r(t), d)$ is the emitted colour from the point $r(t)$ in direction d. To bridge the gap between continuous rays and concrete points that the MLP can process, NeRF modifies the rendering formula to 3.2, which allows us to sample and accumulate concrete 3D points along the ray. In the formula, the δ_i is the distance between adjacent samples, N is the number of sampled points along a ray, and $\hat{C}$ is the final colour of the ray which corresponds to one pixel. For a ray with direction $\mathbf{d}$ and origin $\mathbf{o}$ (which can be computed from camera intrinsic and extrinsic), NeRF firstly samples points within the defined near and far plane along the ray $r(t) = \mathbf{o} + t\mathbf{d}$, where t controls how far the sampled point is from the camera origin. By incrementally varying t from the near plane to the far plane, a multitude of points are sampled, effectively emulating a sense of continuity along the ray:

$$C = \int_{t_n}^{t_f} T(t)\sigma(r(t))c(r(t), d)dt, \text{ where } T(t) = \exp(-\int_{t_n}^{t} \sigma(r(s))ds), \tag{3.1}$$

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i(1 - exp(-\sigma_i\delta_i))\mathbf{c_i}, \text{ where } T_i = exp(-\sum_{j=1}^{i-1} \sigma_i\delta_i). \tag{3.2}$$

We then feed the locations and view directions of these 3D points into the MLP. The MLP in NeRF is trained to accurately predict the volume density (σ) and colour (RGB) of each sampled 3D point (x, y, z) in the scene based on its location and view direction (denoted by $\theta\phi$). The formula 3.3 is a summary of the MLP function, where F is the MLP to predict the colour c_i and the density σ_i given a particular 3D location $\mathbf{x_i}$ and the 2D view direction $\mathbf{d_i}$, γ is the positional encoding function discussed in the next paragraph. In the NeRF implementation, the density is predicted solely based on the 3D position, as it is a physical characteristic that remains constant under different view directions. The colour is predicted by concatenating the 3D position and 2D viewpoint, taking the view-dependent effects such as light and reflection into account:

$$c_i(\mathbf{x_i}, \mathbf{d_i}), \sigma_i(\mathbf{x_i}) = F(\gamma(\mathbf{x_i}), \gamma(\mathbf{d_i})). \tag{3.3}$$

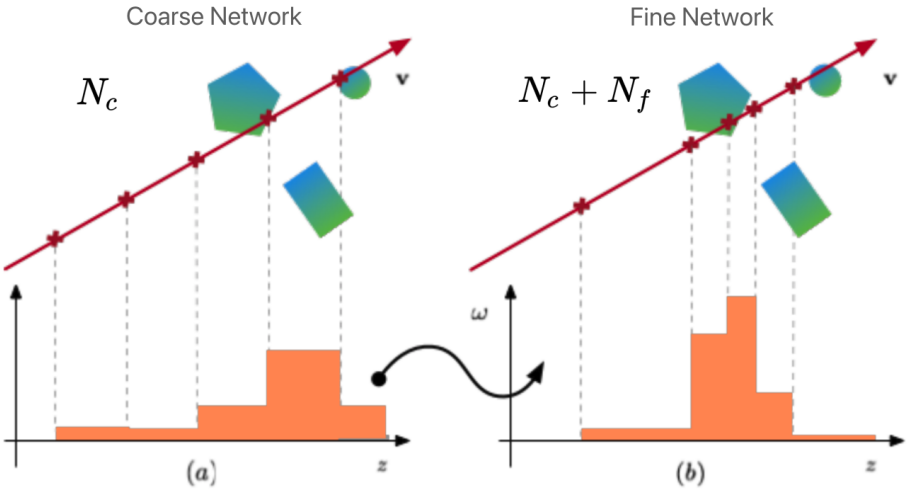


FIGURE 3.7: Hierarchical sampling procedure in NeRF. Figure source: [4]

NeRF further employs several strategies to generate more photorealistic images. The first strategy is a hierarchical sampling procedure as shown in figure 3.7. Instead of uniformly sampling points along rays, which includes free space and occluded points that do not contribute to the ray's colour, NeRF proposes a more efficient sampling procedure which is to simultaneously optimizes two MLPs: a coarse network that samples points uniformly and learns the density, and a fine network that biases the sampling towards

the information segment of the rays (i.e., where the density is high) indicated by the coarse network. Therefore, the fine MLP learns from more informative points, enabling a better representation of the scene's appearance. Fine MLP is used in the final rendering. Another technique utilized in NeRF is positional encoding (γ). It maps the input 5D vector (x, y, z, θ, ϕ) into higher dimensions, addressing the issue of deep networks being biased towards learning lower frequency functions and enhancing the representation of images with high-frequency colour and geometry variations [4]. As shown in formula 3.4, the mapping function applies sine and cosine functions individually to each component of x, y, z, θ and ϕ. Within the formula, these five components are denoted by the variable p. The hyperparameter L determines the frequency of the encoding. By encoding the original 5D input with the frequency function, NeRF enables the MLP to capture and represent fine-grained variations in colour and geometry, resulting in improved fidelity and realism in the rendered images:

$$\gamma(p) = (\sin(2^0\pi p), \cos(2^0\pi p), \ldots, \sin((2^{L-1})\pi p), \cos((2^{L-1})\pi p)). \tag{3.4}$$

Since the classical volume rendering function is intrinsically differential, the parameters in the MLPs can be optimized with gradient descent algorithms to minimize the colour difference between the ground truth and rendered pixels. The loss function incorporates both the coarse and fine MLPs as shown in formula 3.5, where $\mathcal{R}$ refers to a batch of rays, $C(\mathbf{r})$, $\hat{C}_c(\mathbf{r})$ and $\hat{C}_f(\mathbf{r})$ refer to the ground truth, coarse MLP predicted, and fine MLP predicted RGB colours for each ray respectively:

$$L = \sum_{\mathbf{r}\in\mathcal{R}} \left[\left|\left| \hat{C}_c(\mathbf{r}) - C(\mathbf{r}) \right|\right|_2^2 + \left|\left| \hat{C}_f(\mathbf{r}) - C(\mathbf{r}) \right|\right|_2^2 \right]. \tag{3.5}$$

The MLPs in NeRF implicitly store the 3D geometry and view-dependent appearance of the scene, requiring less storage space compared to explicit modelling methods such as discretized voxel grids [4, 30]. Despite its realistic novel view generation capabilities, the original NeRF (which is also called Vanilla NeRF in the context below) has several limitations: 1) It is dedicated to static scenes and cannot be directly applied to dynamic scenes. 2) The MLP only captures the information for one particular scene and requires a long training time. While rendering a new image, a high volume of 3D points needs to

be sampled and the MLP is invoked tremendous times, implying difficulties in deploying to real-world applications where time is critical.

3.4 Dynamic Neural Radiance Fields

While the vanilla NeRF has demonstrated impressive results in generating novel views for static scenes, it cannot be applied to dynamic scenes directly. The main limitation is that the vanilla NeRF assumes the density of the same 3D points and the colour of rays from the same view direction remains constant across frames. In dynamic scenes, such as human portraits, the physical properties and appearance of objects may vary over time, leading to inconsistent input data during training. Such inconsistency leads to biased outputs and blurry regions in the dynamic areas of the scene. To address this limitation, researchers have developed a family of dynamic NeRF models to handle object motions and temporal variations across different time frames [5, 31–36].

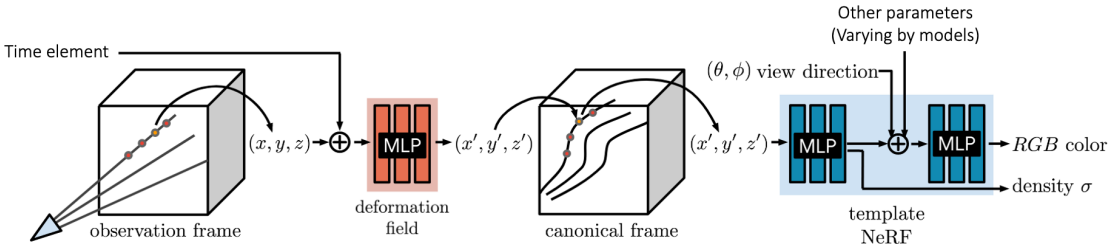


FIGURE 3.8: Two-stage scene modelling, adopted in several dynamic NeRF models. Figure source: [5].

To handle the scene motions over time, D-NeRF [31], NR-NeRF [32], Nerfies [5] and HyperNeRF [33] adopt a two-stage scene learning as illustrated in figure 3.8, which involves a deformation stage and a rendering stage. In the deformation stage, the sampled points are warped into a canonical space using a separate multilayer perceptron (MLP). The canonical space typically represents the scene in the first frame of a sequence of images. The core idea is that points in this canonical space remain constant through time, providing a consistent reference frame for rendering using the rendering MLP (the MLP that outputs the colour and density of a point). To model the dynamic changes over time, D-NeRF [31] introduces an additional parameter t as the input to the deformation MLP, where the first frame corresponds to $t = 0$, and subsequent frames have increasing values of t. The deformation MLP in D-NeRF predicts

a 3D displacement value $\Delta(\mathbf{x})$ that is added to the original point to map it from the deformed space to the consistent canonical space. This process can be summarized as $F_t : (\gamma(\mathbf{x}), \gamma(t)) \rightarrow \Delta\mathbf{x}, F_x : (\gamma(\mathbf{x} + \Delta\mathbf{x}), \gamma(\mathbf{d})) \rightarrow (\mathbf{c}, \sigma)$, where F_t is the deformation MLP, F_x is the rendering MLP, c is the predicted colour, σ is the predicted density, d is the view direction, γ is the positional encoding function (γ) as used in vanilla NeRF.

D-NeRF has proven satisfying results on synthesized scenes created by Blender, but it has not extensively experimented with real-world scenes. NR-NeRF [32], a concurrent work to D-NeRF, successfully applied a similar concept to real-world monocular videos captured by cameras or mobile phones. NR-NeRF utilizes MLPs to transform deformed points into a canonical space by predicting a 3D displacement for each point. Compared to D-NeRF, NR-NeRF employs two networks to infer the deformation: a mask network which predicts a value $M(x) \in [0, 1]$ to determine if a point is static over the scene and does not require deformation, and a deformation MLP which predicts the 3D displacement $\Delta\mathbf{x}$. The final displacement is the product of the outputs from these two networks $M(x) * \Delta\mathbf{x}$. Moreover, compared to the positional encoding of a time parameter in D-NeRF, NR-NeRF conditions the deformation MLP on a learnable per-frame encoding, which is trained alongside the parameters of the mask MLP, deformation MLP and rendering MLP. However, one limitation of NR-NeRF is its assumption of Lambertian materials (surfaces that reflect light uniformly in all view directions). It conditions the colour solely on the 3D position of points and does not incorporate view-dependent effects. This limitation restricts its ability to capture certain visual characteristics that are influenced by the viewing direction.

Another work that adopts a similar approach is Nerfies [5]. Nerfies utilizes a deformation MLP to warp the deformed points into the canonical space, which resembles the deformation model used in D-NeRF. The deformation MLP in Nerfies is conditioned on a per-frame learned latent deformation code, akin to the approach used in NR-NeRF. A notable distinction of Nerfies is that its deformation field produces a dense SE(3) transformation field capturing rotation and translation, rather than 3D displacements ($\Delta\mathbf{x}$). This allows for a simultaneous transformation of clusters of distant points, resulting in improved shape representation. In addition, Nerfies incorporates a per-frame learnable appearance latent code when predicting the colour, which handles appearance differences such as exposure and white balance across frames. Furthermore, Nerfies assumes

large deformations are less likely to occur and employs an elastic regularization to penalise large deformations and thus stabilize the training process. Nerfies also include a background regularization loss to restrict the movement of background points. To solve the issue of overfitting to an under-optimized deformation MLP at the beginning of the training process, Nerfies proposes a coarse-to-fine deformation regularization that varies the frequency of the positional encoding of 3D points. Starting with a low-frequency bias, the frequency is progressively increased, allowing the deformation network to gradually learn how to deform from low resolutions and large motions (such as head rotation) to high resolutions and small dynamics (such as the corners of the mouth).

Despite the impressive results in modelling dynamic scenes, Nerifies exhibits certain limitations, particularly in handling large deformations. The authors of Nerifies acknowledge that the deformation MLP struggles to capture variations in topology, such as mouth opening or paper slicing, and hence proposed HyperNeRF [33] to address such issue. HyperNeRF assumes that the failure lies in the continuity of the MLP to describe the deformation, whereas topological changes are discontinuous. To overcome this limitation, HyperNeRF introduces an additional component called the hyper-field, which is implemented as a separate MLP. This hyper-field maps the input 3D points to a higher-dimensional space. By incorporating the high-dimension feature as an additional input to the rendering MLP, the model becomes capable of generating realistic images with large deformations. HyperNeRF demonstrates remarkable performance in modelling human portraits and provides control over human expressions by manipulating the feature space generated by the hyper-field. However, HyperNeRF lacks explicit control over human expression and head pose, as well as the ability to synthesize unobserved expressions and head poses.

Meanwhile, Neural Scene Flow Field [34], NeRFlow [36] and Dynamic View Synthesis [35] applied a different approach. They predict a 3D motion vector that describes the movement of 3D points between the current time step and neighbouring time steps. Neural Scene Flow Field [34] utilizes a single MLP that incorporates an additional time parameter as input, along with the 3D point and view direction. The MLP predicts four elements: colour and density (similar to the Vanilla NeRF), a scene flow vector $F_i = (f_{i\to i+1}, f_{i\to i-1})$ that represents 3D offset vectors pointing to the positions of the point x in the forward (i+1) and backward (i-1) frames, and a dis-occlusion weight $W_i = (w_{i\to i+1}, w_{i\to i-1})$ to handle the ambiguity that arises when a 3D point becomes

occluded between frames. Neural Scene Flow Field introduces an innovative loss function to ensure that after appending the predicted motion vectors from frame i to frame j to the 3D points in frame i, the rendered image i remains consistent with image j. Moreover, the model incorporates additional loss terms to encourage consistency, such as promoting similarity between the flow vectors $f_{i\to j}$ and $f_{j\to i}$. These loss terms aim to further optimize the model and prevent it from converging to sub-optimal solutions. One limitation of Neural Scene Flow Field is failing to render long or large deformations, which prohibits them from working on complicated scenes.

Dynamic View Synthesis [35] segments the scene into the static background and dynamic area with a segmentation model. They employ two separate models: a static NeRF, which is identical to the vanilla NeRF and is responsible for rendering the static background, and a dynamic NeRF that takes (x, y, z, t) as input to predict the colour and density of the dynamic area, as well as an additional scene flow from the current frame to adjacent frames. During optimization, the static NeRF is optimized using the rendering loss alone, while the dynamic NeRF is optimized using an extra loss, apart from the rendering loss, to enforce consistency in point colour and density over time. This is achieved by mapping the dynamic points using the predicted flow. The dynamic NeRF also incorporates regularization losses, such as encouraging small flows for smoothing, depth supervision using depth prediction models, and ensuring motion consistency between adjacent frames. However, one drawback of dynamic view synthesis is its complex model architecture, which includes two rendering MLPs and several additional steps, such as segmentation and depth supervision. Errors in these intermediate steps could potentially lead to suboptimal results.

NeRFlow [36] introduces a flow field that predicts the motions of the scene through time, where the flow field is a separate MLP that takes the 3D location and time step as inputs and predicts a flow field f representing the movement of each point in space over time. Its radiance field predicts the colour and density based on the 3D location, time step, and view direction. The predicted colour is decomposed into a view-invariant diffuse part (C_{diffuse}) and a view-dependent specular part (C_{specular}) for later loss calculation. To constrain these two MLPs, NeRFlow incorporates several loss terms. The appearance consistency loss ensures that the diffuse colour of the same point remains consistent over time by mapping the point based on the output of the flow field. The density consistency loss enforces the consistency of density for the same point across

different time steps. The motion consistency loss prevents empty points from moving and promotes smoothness in the predicted motion. The rendering loss minimizes the RGB colour difference between the predicted and inferred images. Furthermore, NeRFlow incorporates an optical flow supervision loss to ensure that the flow prediction aligns correctly with the depth map predicted by the depth prediction model. However, NeRFlow did not solve the ambiguities in 3D geometry explicitly hence struggling to model complex real scenes and preserve static backgrounds over time.

In conclusion, several approaches have been proposed to handle dynamic scenes, including the incorporation of deformation fields and scene flow prediction between frames. Some of these models such as Nerfies and HyperNeRF are also experimented with human portraits and have shown promising results in modelling the varying appearance and novel viewpoints of human subjects over time. However, it is important to note that these dynamic NeRFs are limited in their control over other properties of human portraits, such as head pose and facial expression. They are unable to generate portraits with unseen head poses and facial expressions. Nonetheless, these models have laid the groundwork for handling motion in scenes, and their architectures and designed loss functions have been widely adopted in controllable NeRFs as described in Section 3.5 and 3.6.

3.5 Controllable Neural Radiance Fields for Human Body

Our research is closely aligned with the field of rendering human bodies with control over their body pose, size, and camera viewpoint. This field exhibits numerous similarities to our research objectives, particularly in the utilization of models such as SMPL (Skinned Multi-Person Linear model) to represent the 3D human body based on 2D images [37], which is analogous to the use of 3DMMs for representing 3D human faces. Additionally, the model architecture of several controllable NeRFs for bodies bears resemblances to controllable NeRFs for human portraits, where they incorporate a deformation field (similar to the deformation field in D-NeRF, NR-NeRF, Nerfies and HyperNeRF) informed by prior knowledge from 3D meshes.

Animatable NeRF [6] and Neural Actor [38] utilize multi-view videos of individuals and the corresponding human mesh of each frame as inputs to construct controllable NeRF

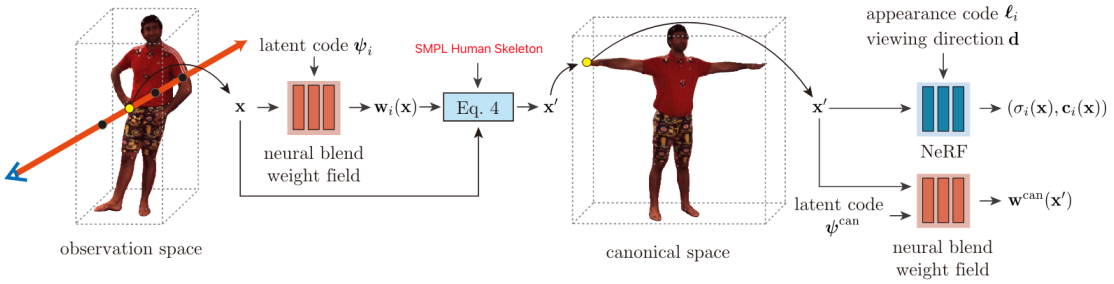


FIGURE 3.9: Animatable NeRF architecture. Figure source: [6].

models. They employ a deformation field based on the human mesh to transform the dynamic human body into a canonical body pose, defined as the T-pose [6] or the Da-pose [38] in the SMPL model. Figure 3.9 presents the model architecture of Animatable NeRF. The transformation of a point in the observation space involves finding the nearest surface point on the SMPL mesh and performing barycentric interpolation of the corresponding vertices of the mesh facet. However, this process may yield imperfect results due to potential estimation errors of the body meshes. To address this, both Animatable NeRF and Neural Actor incorporate a deformation Multilayer Perceptron (MLP) that receives the point 3D position with a per-frame learnable encoding [6] or the SMPL parameters [38] as input to predict a residual deformation, which corrects these errors and enhances the accuracy of the mapping process. To control body pose and generate novel poses, users can manipulate the SMPL parameters to generate a new mesh, thereby controlling the deformation field to achieve desired poses. To further enhance the quality of the rendering outcomes, Neural Actor introduces a technique that integrates texture maps derived from the SMPL model into the NeRF model. This integration effectively addresses uncertainties in dynamic geometry and appearance. Moreover, Neural Actor incorporates a sampling strategy that biases points towards mesh regions, which accelerates both the training and inference processes, improving the efficiency of the model.

In contrast to the deformation field-based approach, Neural Body [39] initiates an alternative methodology to handle body dynamics by establishing connections between points and the body mesh. Instead of using the 3D position as input to the MLP, Neural Body utilizes a latent code that is associated with each vertex on the body mesh. The latent code remains consistent for the same vertex across frames, regardless of its actual 3D location. These latent codes are optimized along with the MLP using gradient descent

during training. The latent code for any 3D point is computed using trilinear interpolation. To address sparse vertices in 3D space, a 3D convolutional network is employed to diffuse the input codes to neighbouring regions. Such methodology ensures that points with similar relationships to their respective meshes (such as those near the hands) share the same latent code. By feeding these latent codes as input, the relationship of points across frames is established.

The above models face certain limitations. Firstly, they are trained on multi-view videos, which can be difficult to obtain due to the need for multiple cameras and precise synchronization. Furthermore, these models assume access to the body mesh and viewpoint information for each frame but do not address the challenge of misalignment between the SMPL coordinate and world coordinate systems. Finally, these models do not incorporate background rendering, which is essential for achieving realistic scenes. Consequently, their applicability in real-world scenarios may be restricted.

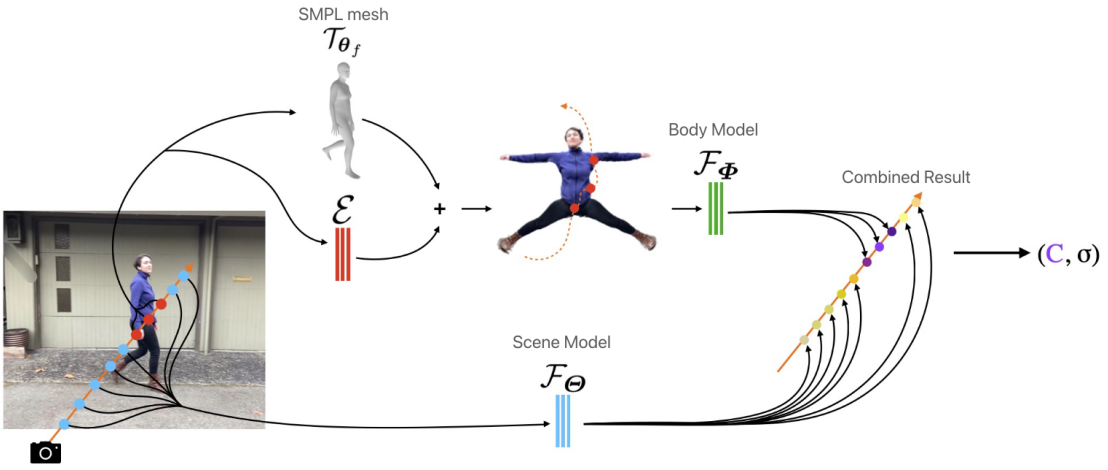


FIGURE 3.10: NeuMan architecture. Figure source: [7].

NeuMan [7] presents an approach for reconstructing both human bodies and the surrounding scene from a single monocular video captured in real-world settings. NeuMan similarly leverages the SMPL model to predict the human body mesh of each video frame captured in unconstrained environments. It employs a segmentation model to distinguish the human from the scene and trains separate models for the body and the scene as presented in figure 3.10. The rendered outputs of these models are combined to create a coherent representation of the body within the scene. To capture fine details not fully represented by the SMPL model, NeuMan incorporates an MLP into the deformation field, similar to Animatable NeRF and Neural Actor. However, since the SMPL

mesh exists in its own coordinate system, NeuMan addresses the alignment between the SMPL space and the world coordinates (the space where NeRF sample points). It first solves the Perspective-n-Point (PnP) problem based on the estimated 3D joints, the projected 2D joints and the camera intrinsic from COLMAP, which aligns the coordinates up to an arbitrary scale. Then it resolves the scale ambiguity by assuming that the human has contact with the ground in at least one frame, finding the scale factor that makes the feet meshes of the SMPL model touch the ground plane. To optimize their models in the presence of ambiguities in monocular videos, NeuMan introduces several loss terms. These include constraining density to be zero for empty space based on a depth map created by fusing the depth map from an off-the-shelf monocular depth estimation model and the depth map from COLMAP, enforcing solid density inside the SMPL mesh and empty density outside the mesh, and applying a hard surface loss to reduce halo artifacts around the human. NeuMan achieves impressive results in rendering human bodies within real-world scenes and requiring less constrained input. However, it assumes at least one contact point with the ground to estimate a scale for coordinates transformation. Additionally, the use of separate models for the human and background may introduce visual inconsistencies, such as unnatural lighting effects between the two.

While these models excel in rendering photo-realistic human bodies with control, it is important to note that the SMPL model used in these approaches does not encompass the intricate details of the human face. Therefore, it is not suitable to directly apply these models to our research objective. Nonetheless, numerous methodologies within this field can be adapted to complement our research efforts.

3.6 Controllable Neural Radiance Fields for Human Portraits

Although several dynamic NeRFs such as NeRFies and HyperNeRF [5, 33] have demonstrated their capability in generating dynamic portraits, and controllable NeRFs for human bodies have proven successful in controlling body poses, they do not adequately support large variations in human portraits, including head rotations, exaggerated expressions while maintaining facial texture details. In order to address the need for comprehensive control over human portraits, a family of controllable NeRFs specifically

designed for this purpose has been proposed [8, 9, 18, 19, 22, 23, 40, 41]. Some of these models utilize 3D Morphable Models (3DMMs) to enable flexible manipulation of portraits [8, 18, 22, 23], while others are based on Generative Adversarial Networks (GANs) [9, 40], or trained from scratch [19, 41].

NerFACE [18] incorporates the pose and expression parameters extracted using the 3DMM face2face model [20] to enable control over the head pose and facial expression of human faces. It takes a monocular video capturing a human portrait as input and separates the background from the portraits, training the NeRF network to learn the dynamic aspects of the portraits. Similar to several dynamic NeRFs and NeRF models for human bodies, NerFACE maps points to a canonical space to handle the moving human head. Instead of employing a deformation MLP, NerFACE uses the 3DMM pose parameters (rotation and translation) to warp points. Additionally, the NeRF MLP in NerFACE conditions on the expression parameters, in addition to the point position and view direction, to learn correct renderings under different expressions. However, NerFACE suffers from several issues. Firstly, conditioning solely on an expression parameter is under-constrained, leading to vague faces under exaggerated expressions. Moreover, the simple deformation based on pose overlooks inaccurate pose estimation and deformations caused by other dynamics. Furthermore, NerFACE assumes a static camera and background, which limits its capability for generating freely controllable views.

FLAME-in-NeRF [22] achieves the manipulation of facial expressions while enabling novel view rendering by leveraging the 3DMM FLAME [2] to extract expression parameters and impose a spatial prior. To transform points into a canonical space, FLAME-in-NeRF adopts a deformation network identical to Nerfies, taking the 3D point position and a per-frame learnable embedding as input. The NeRF MLP simultaneously learns the static background and dynamic facial area. Similarly to NerFACE, FLAME-in-NeRF conditions the NeRF MLP with the expression parameters, but solely for rays within the face. Such an approach addresses the issue of incorrect background appearance when the view remains constant but the expression changes. FLAME-in-NeRF employs the silhouettes provided by FLAME as a spatial prior to identify which rays are related to the face. During the inference of a new portrait, users can modify the expression parameters to alter human facial appearances or change the camera pose to render the face and background from different viewpoints. Nevertheless, the estimated deformation in FLAME-in-NeRF is conditioned on learnt deformation codes that can

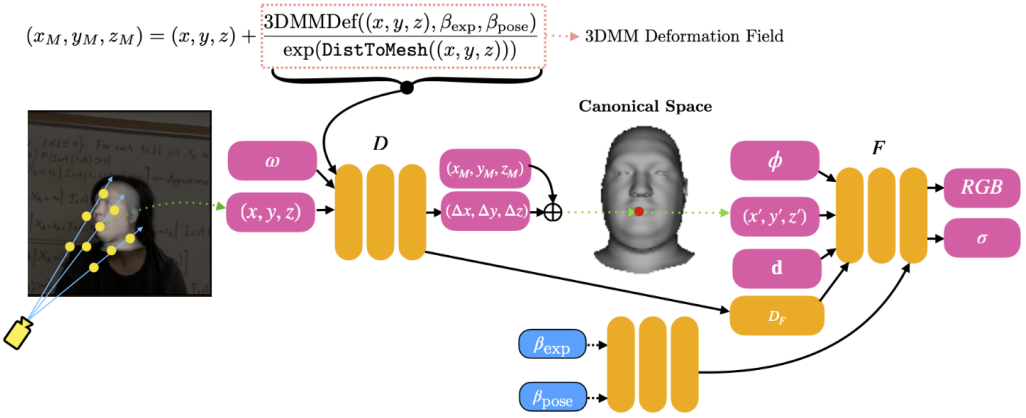


FIGURE 3.11: RigNeRF Architecture. Figure source: [8].

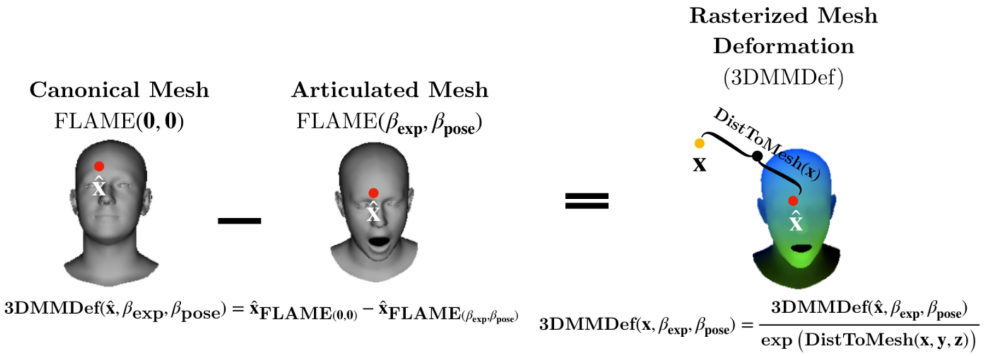


FIGURE 3.12: The 3DMM deformation field presented in RigNeRF. Figure source: [8].

be arbitrary, leading to unnatural traces for new expressions. Furthermore, FLAME-in-NeRF lacks the capability to control head rotations and movement, thereby limiting its overall generality.

The researchers further explored how to apply 3DMM with NeRF and proposed RigNeRF [8]. RigNeRF provides explicit control over the view direction, head pose and facial expression for generated portraits via the 3DMM meshes extracted using the DECA model [24]. Similar to the aforementioned models, RigNeRF deforms the rays into a canonical space before predicting the density and appearance with an MLP as shown in figure 3.11. RigNeRF explicitly defines the canonical space by the frontal head pose and neutral expression of the FLAME model, where both pose and facial expressions are set to zero. In RigNeRF, the deformation is not solely predicted by a single MLP like in FLAME-in-NeRF or based on pose parameters like in NerFACE. Instead, it is determined by the sum of a 3DMM deformation field and a residual deformation estimated by an MLP. The 3DMM deformation for a 3D point $\mathbf{x} = (x, y, z)$ at frame F is

calculated using the observed mesh of frame Mesh_f and the canonical mesh Mesh_{can}, as shown in figure 3.12. This can be described by the equation:

$$\text{3DMMDef}(\mathbf{x},\text{Mesh}_f) = \frac{\text{3DMMDef}(\hat{\mathbf{x}},\text{Mesh}_f)}{\exp(\text{DistToMesh}(\mathbf{x}))},$$

where $\hat{\mathbf{x}}$ represents the closest point to $\mathbf{x}$ on the Mesh_f, and $\text{DistToMesh}(\mathbf{x})$ calculates the distance between $\mathbf{x}$ and $\hat{\mathbf{x}}$. The 3DMM deformation for any point on the mesh, $\hat{\mathbf{x}}$, is determined by the difference between its position in the canonical space and its current articulation:

$$\text{3DMMDef}(\hat{\mathbf{x}},\text{Mesh}_f) = \hat{\mathbf{x}}_{\text{FLAME}(\text{Mesh}_{can})} - \hat{\mathbf{x}}_{\text{FLAME}(\text{Mesh}_f)}.$$

To account for points not captured by the 3DMM, such as those around hair, glasses, and accessories, RigNeRF involves a deformation MLP that conditions on the 3D point position, a per-frame learnable embedding and the 3DMM deformation. This estimated residual deformation not only captures deformations associated with the 3DMM but also model deformations not accounted for by changes in head pose and expression via the learnable deformation codes.

Furthermore, RigNeRF computes the density and appearance of any point based on the 3DMM information. Both expression and head pose parameters, as well as the features extracted from the penultimate layer of the deformation MLP, are fed into the MLP. It is simple to generate portraits with novel head poses and expressions by adjusting the observed mesh through varying expression and pose parameters in DECA. One limitation of RigNeRF is its relatively low resolution (256×256), which prevents capturing fine details of human faces, such as wrinkles and moles. This is because the training time for RigNeRF is long and simply increasing the resolution will lead to slower training. Additionally, RigNeRF does not discuss how they acquire the head meshes within the coordinate where points are sampled, lacking a detailed explanation of the 3DMM deformation computation.

IMAvatar [23] differs from directly applying meshes generated with a pretrained 3DMM as RigNeRF. Instead, it draws inspiration from the FLAME model, which parameterizes facial geometry using shape, pose, and expression components, and employs linear blend skinning to represent varying human portraits. IMAvatar trains a deformation network

from scratch that predicts expression blendshape, pose corrective and the linear blend skinning weights for each point. This enables the transformation of any point, whether on the head surface or not, into a canonical space. In addition, the density of each point is inferred with a per-frame encoding, while the appearance is conditioned on pose and expression parameters, apart from the position and view direction. IMAvatar also includes an optional FLAME loss, which supervises the deformation network by comparing it with the corresponding values of the nearest FLAME vertices. However, IMAvatar does not fully exploit the pretrained 3DMM, besides its utilization in an optional loss. As a result, the deformation process remains under-constrained without utilizing prior information extracted with a 3DMM, leading to a potentially more unstable and biased training process.

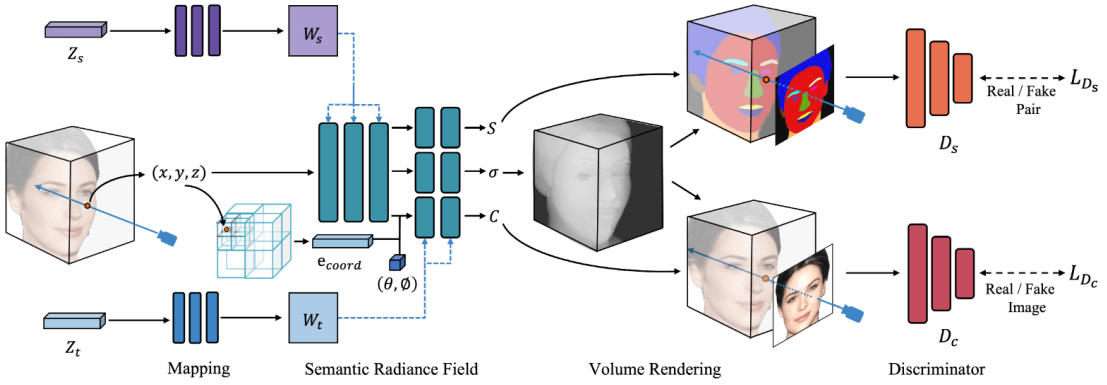


FIGURE 3.13: FENeRF Architecture. Figure source: [9].

Unlike 3DMM-based controllable NeRFs, FENeRF [9] combines a 3D-aware Generative Adversarial Network (GAN) with NeRF to generate view-consistent and locally-editable portrait images. As displayed in figure 3.13, the architecture of FENeRF consists of a conditional generator responsible for predicting the colour and density of each 3D point (akin to the Multi-layer Perceptron in the vanilla NeRF), where the final image is created by applying volumetric rendering, and a discriminator to discern the realism of the rendered images. To enable free control of the portraits, FENeRF introduces two latent codes as conditioning inputs to the generator: the shape code controls the geometry of portraits, and the texture code governs their appearance. The generator takes these latent codes, along with the point position, view direction, and a feature vector from a learnable 3D grid which compensates high-frequency image details as input, yielding colour, density, and a semantic mask. The semantic mask encourages alignment between the appearance and semantics of the face, facilitating better facial control. FENeRF

utilizes a CNN-based discriminator to assess the fidelity of the rendered portrait and trains the generator to deceive the discriminator, enhancing the overall realism of the generated images.

3DFaceShop [40] also adopts a 3D-aware GAN and integrates a 3D Morphable Model (3DMM) to effectively govern facial shape, expression, and lighting, alongside the NeRF technique to ensure multi-view consistency and deliver photo-realistic results. In comparison to FeNeRF, 3DFaceShop employs a distinct strategy by utilizing the identity, expression, and illumination parameters extracted from a pretrained 3DMM as conditions for the generator, rather than relying on latent codes that are trained concurrently with other parameters as in FeNeRF. This design choice reduces the training complexity and capitalizes on the inherent nature of the 3DMM to disentangle control over various facial attributes. The generator in 3DFaceShop receives the latent identity, expression, and illumination codes, in addition to the standard 3D point and view direction inputs, to predict the colour and density of each point. The discriminator provides an adversarial loss, which compels the generator to produce realistic portrait images while maintaining precise control over various facial properties. To further enhance the disentanglement of the 3DMM parameters and improve the control of generated portraits, 3DFaceShop developed a training strategy that constructs contrastive image pairs that exhibit differences in partial latent segments, and enforces the consistency for the attributes that share identical latent codes. Such a step makes sure that each latent code from 3DMM will be independent of others, leading to better local control without impacting other facial properties.

However, FENeRF and 3DFaceShop fail to address our research question since they are trained on a large face dataset to generate realistic faces without explicit control over the identity being generated. Users can manipulate the identity latent code to synthesize different portraits, but cannot specify one known subject. Additionally, they assume a consistent view or static background or empty background and do not have the capability to generate portraits from novel viewpoints.

CoNeRF [19] introduces further enhancements to the expression control. It subdivides facial expressions into individual attributes, where an attribute may refer to a smile or a frown for the mouth, or open or closed for the eyes. Users have the flexibility to define a custom set of attributes and sparsely annotate the training data to indicate the regions

affected by each attribute. Each attribute is then considered as a latent variable that is regressed by the neural network. CoNeRF applies a structure akin to HyperNeRF, which includes a hyperspace field to handle these attribute latent variables. Additionally, to enrich the training images with more annotations, CoNeRF trains regressors to predict the attribute and mask for each image with the sparsely annotated images as the training dataset. During synthesis, users can manipulate the values of each attribute to generate new expressions. Besides, CoNeRF sets a learning objective to incorporate explicit control into the expression representation. This objective encourages each image to represent a distinct value of the attribute encodings, facilitating more fine-grained control over expressions. CoNeRF introduces additional modules, resulting in more complex networks, longer training and inference time. Furthermore, the method requires annotated images to specify the regions affected by each attribute, which can pose challenges in practical applications. CoNeRF primarily focuses on expressions and does not generate new head poses or the background under novel viewpoints.

CoNFies [41] represents a notable advancement over CoNeRF by addressing the laborious and time-consuming process of manual annotations by introducing an automatic annotation mechanism. It leverages an off-the-shelf facial action recognition system (OpenFace [42]) to characterize the facial expressions of each frame as a combination of facial action units (AUs) and their corresponding intensities. Each facial AU corresponds to a specific local landmark on the face, and the intensity describes the degree of landmark movement, similar to the facial attribute annotations employed in CoNeRF. Following the underlying architecture of CoNeRF, CoNFies constructs a face hyper-space that takes into account the 3D point, view direction, semantic masks, and AU intensities. This integration enables control over generating novel views for portrait rendering and exploring unseen expression combinations. By eliminating the need for manual annotation, CoNFies significantly improves the applicability of the approach. However, CoNFies still face certain limitations, including the absence of head pose control, static background representation, and the complexity of the model.

In short summary, compared to other controllable portrait generation methods like 3D Morphable Models (3DMM), Neural Radiance Fields can capture fine details in portrait images beyond just the facial area, including hair details and mouth interior, while classical methods often struggle to obtain such details due to a lack of 3D geometry. Additionally, NeRF models do not suffer from discretization artifacts found in voxel grids,

as they operate on continuous implicit representations. Furthermore, they can account for view-dependent effects, such as lighting and reflections. To enable precise control over various properties in human portraits, existing models combine NeRF with other techniques such as 3DMM, conditional 3D GAN, facial action recognition systems and so on, leveraging their inherent ability to separate each property of portrait into distinct codes, allowing for better control over individual facial components. However, existing models still exhibit certain limitations. As shown in table 3.1, some models may not adequately handle background information, leading to incomplete results. Other approaches may require additional annotations, making the process more complex. Additionally, some models may struggle to accurately identify identities, resulting in less ideal rendering outcomes. Long training and rendering times can also impede these models from being applied in real-world scenarios.

Our research aims to address these issues by extending existing models to render arbitrary casually taken portrait videos within any background. Additionally, we focus on improving training efficiency to ensure that these models can be applied in real-world scenarios without significant time overhead.

Models	Novel view	Background	Novel head pose	Novel expression	Explicit Control
NerFACE		✓	✓	✓	✓
FLAME-in-NeRF	✓	✓		✓	✓
RigNeRF	✓	✓	✓	✓	✓
IMAvatar			✓	✓	✓
FENeRF	✓		✓	✓	
3DFaceShop	✓		✓	✓	
CoNeRF	✓	✓		✓	✓
CoNFies	✓	✓		✓	✓

TABLE 3.1: Summary of the capabilities and limitations of each controllable NeRF for human portraits.

3.7 Efficient Neural Radiance Fields

The vanilla NeRF and the previous variations designed to handle scene dynamics have a significant drawback of requiring substantial computing power for both training and rendering processes. Training a NeRF model for static scenes with a high-performance GPU can take approximately two days, and more complex models may take even longer. Since each NeRF model needs to be trained for a specific scene, the extended training time hinders the deployment of NeRF-based models in real-world applications where

shorter or even real-time training is expected. Furthermore, rendering each frame requires querying the MLP hundreds of times and takes around half a minute for each frame. Rendering high-resolution videos with commercial hardware could take hours.

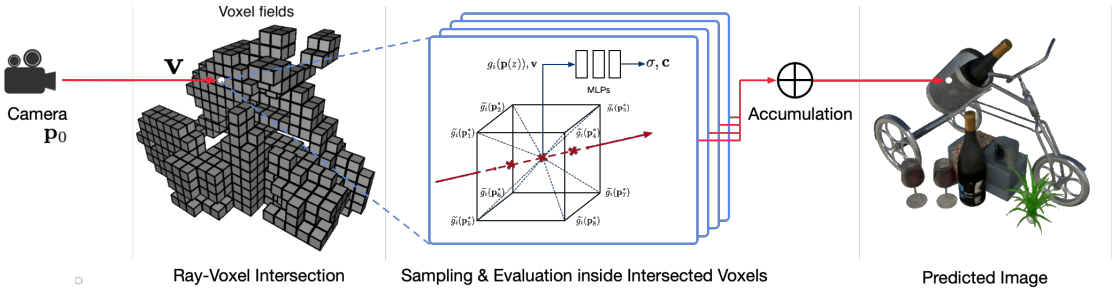


FIGURE 3.14: Various models adopt Voxel Fields to accelerate the training and rendering efficiency of NeRF. Figure source: [10].

To address these challenges, NeRF models with higher efficiency and faster rendering capabilities have been introduced. Some models make trade-offs in storage space to achieve faster training and inference. For example, several approaches adopt voxel fields [10, 30, 43–45] to store the information of a 3D scene (figure 3.14). NSVF [10] recognizes the vanilla NeRF's lengthy training time is primarily due to using a large Multilayer Perceptron (MLP) as an implicit function to model the entire scene. To obtain the colour of each pixel, the vanilla NeRF requires invoking the large MLP hundreds of times, resulting in time-consuming computations. In contrast, NSVF comprises a set of sparse voxel-bounded implicit fields $\mathbf{V} = \{V_1, ..., V_k\}$, where each vertex V_n of the voxel stores a feature vector that serves as trainable parameters. When predicting the colour and density of a point, the model first aggregates the feature embeddings from the eight nearest vertices through trilinear interpolation. The aggregated features are then further processed by a shallow multilayer perceptron to derive geometry and view-dependent appearance. The hierarchical octree structure organizes the voxel field, enabling faster querying during the training and rendering phases. Compared to NeRF, NSVF stores semantic vectors within the voxel field, embedding region-specific information such as geometry, materials, and colour. This information can be directly queried and processed by a lightweight MLP, which is generally more than ten times smaller than the original MLP used in NeRF. As a result, NSVF maintains image quality similar to the original NeRF while significantly reducing the time required to compute density and colour for each 3D point. Additionally, NSVF applies an efficient self-pruning mechanism during training, removing non-essential voxels based on the coarse geometry described using the model's predictions of density. During inference of unseen views, the model directly

samples points based on the vertices in the voxel, avoiding the need for a coarse-to-fine sampling strategy. These optimizations effectively reduce the number of points sampled and further contribute to shorter training times.

DVGO (Direct Voxel Grid Optimization) [43] builds upon the concept of voxel fields and introduces several improvements to enhance convergence speed and output quality. DVGO utilizes two voxel fields, one storing the density representing the scene geometry and the other storing feature vectors processed by a shallow network to model complex view-dependent appearance. Two key improvements introduced by DVGO are the post-activation interpolation on voxel density and the use of priors to address suboptimal geometry solutions. The first improvement, post-activation interpolation on voxel density, involves applying the sigmoid activation function, which enforces density values to lie within the range of 0 to 1, after trilinear interpolation of the density voxel grid. This approach produces sharp surfaces even at lower grid resolutions, resulting in higher-quality output and preventing surfaces from appearing overly smooth. The second improvement tackles the suboptimal geometry solutions, where DVGO proposes two priors. Firstly, the density voxel grid is initialized to yield opacities very close to zero everywhere. This initialization prevents the geometry solutions from being biased towards the cameras' near planes. Secondly, DVGO employs a lower learning rate for voxels that are visible to fewer views. This helps avoid the allocation of redundant voxels that are solely meant to explain observations from a small number of views, leading to a more efficient and effective optimization process. By leveraging these tricks, DVGO achieves better geometry solutions and further improves the overall performance of the model.

SNeRG [30] similarly adopts a voxel field to store the scene information. Instead of a simple feature vector, each voxel grid in SNeRG contains density, diffuse colour and a learned feature vector that is specifically designed to handle view-dependent effects. To render a pixel, SNeRG first samples points in space and retrieves the density, diffuse colour, and feature vector of each point by querying the voxel field and performing trilinear interpolation. To infer view-dependent colours, SNeRG accumulates the diffuse colours and feature vectors along the ray based on the densities. Then the accumulated feature vector and the positional embedded view direction are passed through a lightweight MLP to generate a view-dependent residual colour. The eventual colour is the sum of the accumulated diffuse colour and the residual colour. The significant

advantage of SNeRG is that the lightweight MLP only runs once per pixel with the accumulated feature vector, unlike vanilla NeRF, NSVF, and DVGO, where the MLP is executed per 3D point. This optimization results in significantly faster rendering speeds during both training and testing stages. SNeRG experimentally proved that such a method can accelerate NeRF by three orders of magnitude without losing too much quality and detail.

Meanwhile, E-NeRF [45], DONeRF [46] and AdaNeRF [47] improve the sampling strategy in the vanilla NeRF, leading to remarkable acceleration during both training and testing phases. E-NeRF [45] observed that when sampling uniformly, only around 20% of the sampled points are valid (with non-zero density) and 10% are pivotal (with density higher than a preset threshold), where the remaining points are processed by the MLP but do not contribute to the final colour. E-NeRF maintains a density voxel field that stores the geometry of the scene. E-NeRF observed that the self-pruning mechanism in NSVF may lead to adverse effects when removal is wrong, hence the density voxel does not have any pruning. E-NeRF updates the density voxel in an online manner with momentum to stabilize the training process. During the coarse stage, the MLP only infers valid samples based on the voxel information. For the fine stage which samples more points to produce a higher quality image, E-NeRF additionally introduces the pivotal sampling strategy which focuses on inferring the nearby area of pivotal points only. By doing so, the number of sampled points is substantially reduced while maintaining comparable accuracy. E-NeRF utilizes a novel tree-based data structure to more efficiently represent 3D scenes. By refining the sampling strategy, E-NeRF achieves efficient model training and quick inference for novel views. These improvements make E-NeRF an efficient and practical solution for rendering high-quality scenes.

DONeRF [46] proposes a compact sampling strategy to render pixels effectively using points around surfaces only. It adopts a dual network design, consisting of a sampling oracle network and a shading network. The sampling oracle network takes a ray as input and predicts the likelihood that a particular segmentation of the ray will improve image quality or not. The network is modelled as a multiclass classification task, where each class corresponds to a discretized segment along the ray. This oracle network is called per ray to guide the selection of points that need to be fed into the later shading network, effectively reducing the number of samples required. The second network is

a shading network that utilizes NeRF-like raymarching accumulation to generate RGB output, allowing for accurate rendering of scenes.

Analogous to DONeRF, AdaNeRF [47] also adopts a dual-network architecture. AdaNeRF includes a sampling network and a shading network to efficiently generate the final image while minimizing the number of network evaluations. The sampling network takes the ray origin and direction as input and outputs a vector of the importance of points along the ray. By leveraging the sampling network, AdaNeRF reduces the number of network evaluations required to generate the final image. The shading network takes the predictions from the sampling network and outputs the density and colour of the points with the highest contribution. The loss function in AdaNeRF includes an additional L1 loss term for the sampling network to force the majority of predictions towards 0 and encourage sparsity.

DONeRF and AdaNeRF significantly reduce the inference costs compared to the vanilla NeRF with the use of a per-ray evaluated sampling network. In addition, compared with the above voxel-field-based methods, they do not require additional memory for explicit caching acceleration structures, making it a memory-efficient alternative.

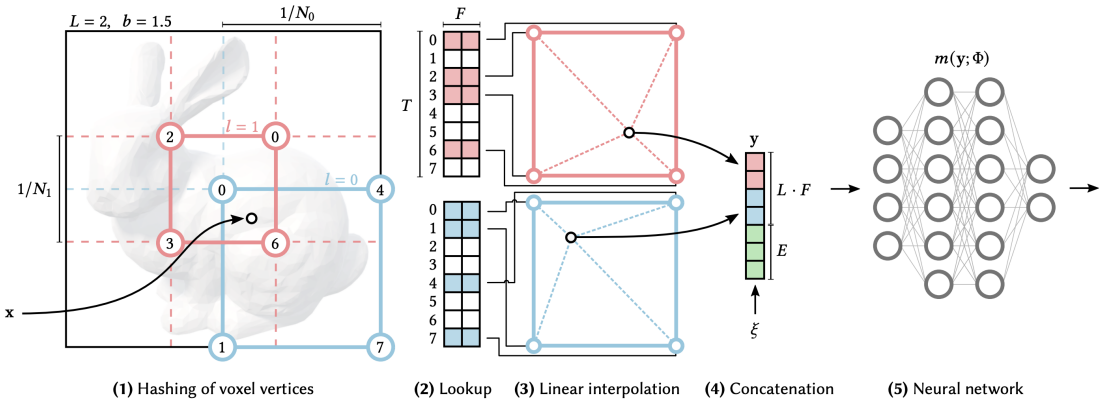


FIGURE 3.15: InstantNGP multi-resolution hashing encoding architecture. Figure source: [11].

InstantNGP [11] presents an innovative approach that introduces a novel multi-resolution hash encoding method, aimed at replacing the original non-trainable trigonometric functions-based positional encoding. This substitution enables the utilization of a more compact network architecture without compromising the output quality, thus leading to a significant reduction in the number of calculations performed by the MLP. As detailed in figure 3.15, the proposed hash encoding method facilitates the mapping of 3D points into a higher-dimensional space, where the parameters within the encoding are trainable

alongside the parameters in the MLP through backpropagation. The spatial hash table is structured into L levels, each comprising up to T feature vectors of dimensionality F. Each level operates independently and conceptually stores feature vectors at the vertices of a grid. The values of L, T, and F can be fine-tuned to strike a balance between computational efficiency, performance, and memory utilization. To infer the colour and density of a point, its 3D position X is mapped to corresponding voxels (d-dimensional cells) at each level using a spatial hash function. Subsequently, each d-dimensional vertex is associated with one of the T feature vectors from the respective level. As a result, d feature vectors are obtained for each layer, each of which corresponds to the vertices of the respective voxel. Then, linear interpolation of these d vectors is applied to derive the final vector representing the layer. The concatenation of the final vectors from each layer, together with the view direction, yields the vector that serves as input to the multilayer perceptron (MLP). From this point onwards, the procedure follows a similar methodology to that of the original NeRF. By combining different resolutions, the model can semantically encode each 3D point to generate images preserving high-frequency details. Notably, this methodology does not depend on progressive pruning during training or any prior knowledge of the scene's geometry. InstantNGP employs the hash function resulting in a reduction in memory access operations and inference speed while preserving the quality of fine details.

In conclusion, various methodologies are proposed to handle the long training and rendering time in the original NeRF, including the usage of voxel-field or altering the sampling strategy. However, as these efficient NeRFs are established based on the vanilla NeRF, they can only be applied to static scenes. In dynamic scenes, the properties (e.g. density, appearance) of 3D points may alter over time. Simply querying the voxel grids or sampling network without deformation may result in misalignment, indicating extra efforts are required to suit dynamic scenes and objects.

3.8 Efficient Dynamic NeRF

Dynamic NeRF models typically exhibit more complex architectures with a greater number of parameters compared to the vanilla NeRF, resulting in a lengthier training time. To enhance their training efficiency, several methodologies have been devised by incorporating efficient NeRF architectures into dynamic NeRFs. A common approach

involves the replacement of the original rendering Multi-layer Perceptron (MLP) with optimized data structures such as voxel grids [43] or hash encoding [11]. Furthermore, several models [12, 48] apply these efficient NeRF architectures within the deformation module with adjustments to deform rays faster. These integrations aim to strike a balance between effectively handling dynamic motions and maintaining a reasonable training speed.

DeVRF [48] accelerates the learning of dynamic scenes by incorporating explicit and discrete voxel-based representations for modelling both the canonical space and the deformation. It builds upon the foundation of DVGO [43] and introduces two voxel fields. The first voxel field models properties such as the density and colour of the canonical space scene and is represented by a density grid, a colour grid and a shallow Multi-Layer Perceptron (MLP) to capture view-dependent information; The second voxel field is a 4D grid, with three spatial dimensions and an additional time frame dimension. It is responsible for modelling how each dynamic point is transformed into the canonical space by storing a 3D displacement vector in each grid. DeVRF introduces a static-to-dynamic learning strategy. It initially films a static scene with a moving monocular camera. The static scene is treated as the canonical space. The canonical voxel field is optimized first using these static frames. Subsequently, a few fixed cameras are used to capture the motions from multi-views, and the entire model is trained using the later dynamic frames. This strategy begins by training the canonical voxel field using static frames, a task that is less ill-posed and comparatively less complex. This phase aims to establish a correct canonical voxel grid. Consequently, when the entire model is trained with dynamic frames, the primary focus shifts to refining the deformation voxel field and can effectively mitigate the risk of overfitting due to a large number of deformation field parameters.

Analogous to DeVRF, NDVG [12] adopts voxel-grid representations to manage deformation calculations and colour and density inference as shown in figure 3.16. However, NDVG claims that it is inefficient to directly store the offsets in a 4D grid. NDVG employs a hybrid deformation field comprising a 3D feature grid and a lightweight Multi-Layer Perceptron (MLP). Each feature grid contains learnable feature vectors that store semantic information regarding the deformation of rays. For each point, these grids enable the acquisition of a feature through trilinear interpolation. The interpolated feature, combined with a time encoding factor t, is then decoded by the lightweight MLP

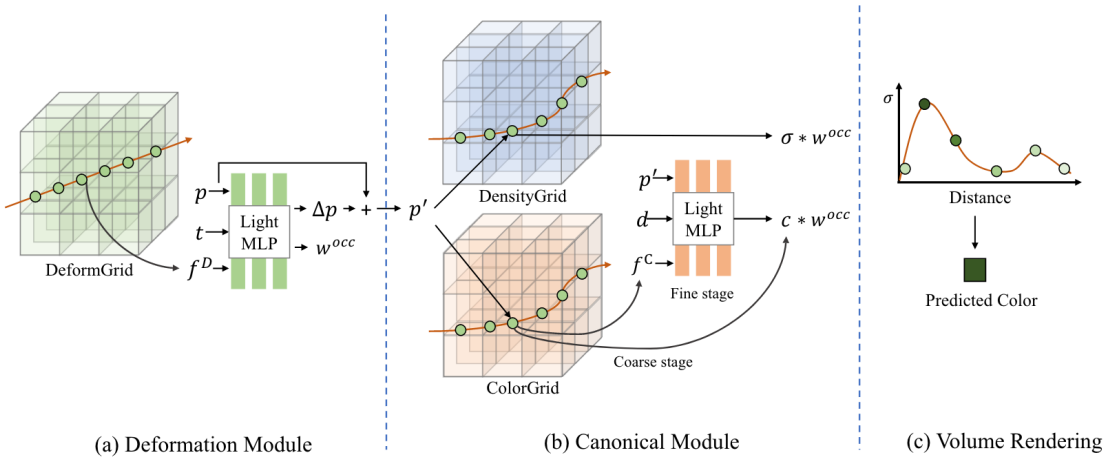


FIGURE 3.16: NDVG architecture. Figure source: [12].

to generate a displacement vector that maps the point to its corresponding location in the canonical space. The canonical voxel grid employed in NDVG is identical to the one utilized in DeVRF and DVGO, containing a density grid, a colour feature grid and a lightweight MLP. Moreover, NDVG addresses a notable issue associated with canonical scene-based methods. These methods assume that "empty" points (i.e., non-object points) in the space remain static and have no deformation. However, a challenge arises when a 3D point is considered an empty point at a given time t but is occupied by an object point in the canonical space, leading to the incorrect rendering of the empty point and occlusion. To mitigate this problem, NDVG introduces an occlusion factor as an additional output of the deformation MLP. For empty points, the occlusion factor is anticipated to be zero. This factor is later multiplied with the colour and density values to obtain a new colour and density, such that the incorrect non-zero colour and density will be neglected during the volumetric rendering, effectively resolving occlusion-related issues to ensure more accurate scene renderings. NDVG optimizes the entire model in an end-to-end manner to minimize the rendering loss.

DeVRF and NDVG employ voxel grids to expedite the learning process for dynamic scenes, primarily catering to minor motions. However, they struggle to model extensive deformations, such as those encountered in the human body and portrait settings, due to solely conditioning on time encoding without additional guidance. To model dynamic human bodies and portraits, it has been suggested from the preceding sections that incorporating pretrained models like SMPL and 3DMMs as priors could easier and more accurately calculate the deformation. Several approaches have explored the integration of acceleration techniques into these models. For example, InstantAvatar [49] introduces

a training acceleration method tailored for human bodies. This method employs a three-stage process with an empty point skipping technique. It firstly transforms points from the deformed space into a normalized space based on the rotation and translation parameters predicted by SMPL, where the global orientation and translation of human poses are factored out. It then filters empty points in this normalized space using an occupancy grid trained to store occupied points across different body poses. Finally, the remaining points are further deformed into the canonical space and processed through an Instant-NGP-based module to predict colour and density. Since the occupancy grid captures the union of occupied space in all frames after inverting translation and rotation, it is ensured that empty points are trimmed without compromising any content points. The incorporation of InstantNGP further fastens the learning process for the appearance and density of the canonical human body.

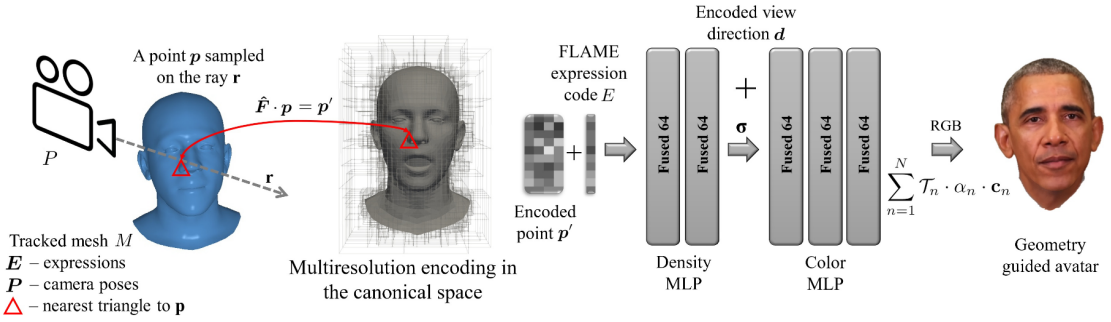


FIGURE 3.17: INSTA architecture. Figure source: [13].

Furthermore, INSTA [13], akin to prior research, integrates a 3D Morphable Face Model (3DMM) to guide deformations (figure 3.17). It speeds up the rendering of dynamic human heads by reducing the model complexity and leveraging InstantNGP. INSTA uses FLAME to track the mesh for each frame and it performs point deformations solely based on the FLAME meshes without additional modules, which significantly reduces computational complexity. To refine deformation calculation accuracy, INSTA employs the nearest triangle search which imputes deformation by referencing the three closest vertices, and then deduces a rotation matrix based on the corresponding tangent, bitangent and normal vectors associated with the triangle. Additionally, recognizing the potential variance in triangle scales across deformed and canonical spaces, INSTA derives an isotropic scaling factor calculated by weighing the relative change in the surface area of a triangle compared to its canonical counterpart. For real-time rendering and optimization of the model, INSTA utilizes a classical bounding volume hierarchy (BVH) [50] which is a tree-based structure to expedite the nearest triangle search, thus

optimizing deformation computations. In INSTA, the canonical space is encapsulated by the InstantNGP model, which also plays an important role in boosting the system's overall efficiency.

These models demonstrate the viability of incorporating the efficient NeRF architecture into dynamic models, thereby accelerating the training process without substantially compromising rendering quality. While InstantAvatar and INSTA offer commendable training efficiency, they fall short of our research objectives, particularly our aspiration to capture the entire human portrait with any real-world background. InstantAvatar yields blurry results in the facial region, attributable to the coarse facial modelling by SMPL. Both models neglect the background scene during rendering, focusing solely on the body or head. This omission prevents the contextual placement of these figures in a realistic environment. Our research aims to develop a model that addresses these limitations and maintains efficient training.

3.9 Review Summary

In our review, numerous human portrait image generation approaches have been studied previously, including the 3DMMs and GANs. However, these methods suffer from several limitations such as explicit modelling tends to consume considerable memory, and lack of 3D information exhibiting artifacts while varying the viewpoint. On the other hand, NeRF proposed a cutting-edge technology based on the classical volume rendering technique to generate high-quality novel-view images, by encapsulating the scene into a single MLP implicitly. However, the original NeRF is constrained to static scenes and suffers from long training time. Many subsequent advancements have amplified the efficacy and adaptability of NeRF across diverse contexts: numerous efficient NeRFs adopt extra data structures or modules, enabling fast model training and real-time image rendering; multiple dynamic NeRFs can handle the scene motions; controllable NeRFs broaden the capabilities of dynamic NeRFs to not only render motions in the scene, but also can control the human body and portraits to generate unseen motions in the training dataset; several approaches have also combined efficient NeRFs and dynamic NeRFs to enable high-efficiency training while ensuring the rendering of high fidelity images with dynamics elements or enabling the generation of new human portraits.

Despite these advances in NeRF, there remains a discernible gap: to the best of our knowledge, existing controllable NeRFs can only produce low-resolution images and fail to handle large rigid deformations under novel viewpoints when considering the background scene. One plausible explanation is the inherent challenge faced by a simple MLP to infer all rigid deformations. Even though certain models incorporate priors like 3DMM, they do not position the meshes and the background within a coherent space. This misalignment can precipitate ambiguous complications when optimizing the entire model. Additionally, the inability to seamlessly integrate efficient NeRFs might explain their struggles with rendering high-resolution images given the protracted training durations. Our research endeavours to surmount these challenges, aiming to develop controllable NeRF models that not only render dynamic human portraits but also ensure expedited training.

Chapter 4

Methodology

In this research, we introduce a methodology for rendering novel portrait images with a high degree of editability. Drawing inspiration from existing works and integrating with our own innovations, we propose a model, named 3ENeRF (3DMM-based Efficient NeRF), which leverages a combination of a 3DMM and a boosting technique to accelerate the training process. In essence, our model harnesses an available pretrained 3DMM to extract the explicit 3D modelling of human portraits as prior information, assisting the NeRF model to autonomously learn the dynamics, geometry, and appearance of human portraits from scratch. Such an approach only requires simple monocular videos captured using a phone camera as the training data. Once trained, our model empowers users to exercise control over various aspects of the portrait, including viewpoint, facial expression, and head pose. Such versatility facilitates activities like mimicking the behaviours of other people or generating previously unseen facial expressions.

A notable breakthrough of our work lies in the approach of portrait data collection and processing. We address a significant challenge, which is the consistency between the off-the-shelf 3DMM coordinate and the world coordinate within which NeRF operates. While certain previous models have incorporated 3DMMs as priors, none have described their technique to place the head mesh in the world coordinate. For instance, NerFace extracts expression and pose parameters from 3DMM. However, it only applies the pose parameters to transform all points by its inverse and utilizes the expression parameters to condition the MLP to change the appearance under different facial expressions. These simple methods result in the failure to produce multi-view images and produce artifacts

under substantial pose or exaggerated expression conditions. RigNeRF leverages the 3DMM mesh to compute deformations but does not discuss how they obtain the head mesh in the world coordinate in the paper. To fill this knowledge gap, we discuss in detail our methodology for converting all meshes to the world coordinate, powered by our designed data collection requirements and data processing flow.

In this section, we delve into our proposed method for rendering novel portrait images from captured videos that can seamlessly integrate the portrait and background. Our exposition begins with an overview of the overarching structure of our final model and subsequently evaluates each constituent part, elucidating its purpose and underlying rationale. Crucially, we elaborate on how we confront a key challenge involving the non-correspondence between 3DMM-coordinate meshes and world-coordinate meshes in detail. This issue is systematically addressed during the data processing section, where we formulate the problem and elaborate on our approach to tackling it.

4.1 Proposed model

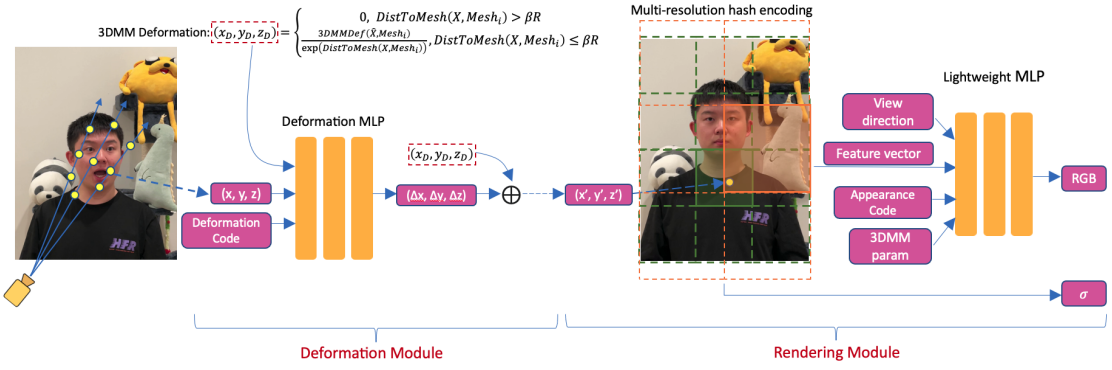


FIGURE 4.1: Proposed model Architecture. The model is made up of two modules: a deformation module and a rendering module. The deformation module includes a 3DMM Deformation and an MLP (explained in section 4.1.1), and the rendering module includes a multi-resolution hash encoding and a lightweight MLP (explained in section 4.1.2).

Figure 4.1 demonstrates an overview of the architecture of our proposed model. This model is structured into two pivotal modules: the deformation module and the rendering module, each encompassing several constituent components. Our design draws inspiration from several preceding methodologies, notably those tailored to address scene dynamics with NeRF, such as D-NeRF [31], Nerfies [5], HyperNeRF [33], and RigNeRF

[8]. The utilization of such architecture, particularly the coupling of the deformation and rendering modules, has consistently demonstrated its ability to resolve the ill-posed challenges linked to learning dynamic scenes. The following sections will comprehensively explain the functionalities of each module.

4.1.1 Deformation Module

The deformation module resolves the critical issue associated with the varying physical attributes of 3D points, encompassing their geometry and appearance, which can undergo temporal differences and hence produce inconsistency for the rendering MLP (the MLP to predict the colour and density of each point). The deformation module computes transformation values that realign dynamic points within a canonical space that remains temporally invariant. Such canonical space establishes a uniform reference frame for the rendering MLP training process.

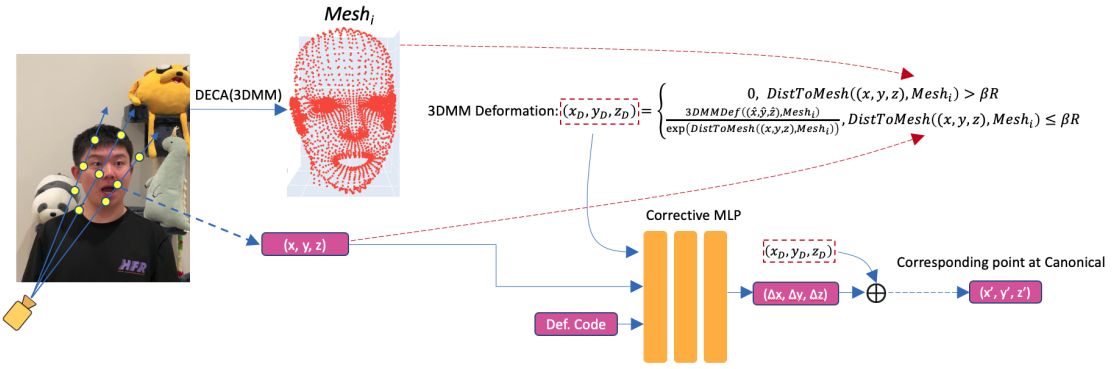


FIGURE 4.2: Our deformation module in details.

As depicted in figure 4.2, the fundamental concept underlying our deformation module revolves around incorporating head meshes generated by a pre-trained 3DMM, particularly where we employ one of the state-of-the-art 3DMM models DECA [24], to calculate a 3DMM deformation that transforms points to a canonical space. Additionally, we introduce a residual block modelled as a single MLP to infer deformations that cannot be obtained through the 3DMM deformation alone.

Our 3DMM deformation is designed based on RigNeRF, which deforms 3D points according to their closest vertices on the predicted head mesh. For each frame with the current deformed head mesh Mesh_D and a canonical reference mesh Mesh_{can} (we will

explain how we generate a canonical mesh in section 4.2), RigNeRF defines its 3DMM deformation (3DMMDef) of each point $\mathbf{x} = (x, y, z)$ as follows:

$$\text{3DMMDef}(\mathbf{x},\text{Mesh}_D) = \frac{\text{3DMMDef}(\hat{\mathbf{x}},\text{Mesh}_D)}{\exp(\text{DistToMesh}(\mathbf{x}))},$$

where $\hat{\mathbf{x}}$ represents the closest vertex to $\mathbf{x}$ on the Mesh_D. The determination of this vertex is based on the Euclidean distance computed between the point $\mathbf{x}$ and all vertices within the Mesh_D. DistToMesh($\mathbf{x}$) refers to the distance between $\mathbf{x}$ and $\hat{\mathbf{x}}$. The 3DMM deformation for any point on the mesh, $\hat{\mathbf{x}}$, is determined by the difference between its position in the canonical space and its current articulation:

$$\text{3DMMDef}(\hat{\mathbf{x}},\text{Mesh}_D) = \hat{\mathbf{x}}_{\text{DECA}(\text{Mesh}_{can})} - \hat{\mathbf{x}}_{\text{DECA}(\text{Mesh}_D)}.$$

RigNeRF undertakes deformation for all points, overlooking the need for certain points to remain static, such as background points. The fact that our meshes are within the world coordinate system affords us the advantage of constraining the deformation range within a certain radius in accordance with the spatial position of meshes. We introduce a radius factor denoted as β, which governs which points are permissible for deformation based on their relative positions to the vertices in the head mesh. For each frame, we first compute the centre of the current deformed mesh. Leveraging the approximation of the head as a rough sphere, we determine the maximum distance from the centre to each vertex, yielding the maximum radius. This radius provides us with a rough guide to the scale of the mesh. Our approach then revolves around selectively deforming points based on their proximity to the mesh, regulated by the parameter β. More explicitly, we exclusively deform points that lie within a certain range relative to the mesh. For all points where DistToMesh exceeds β times the maximum radius (i.e., βR), we nullify their deformation values by setting them to 0. In our model, we set β to 0.5 to take the potential imperfect position of meshes and the sparsity of vertices into consideration. This strategic approach effectively precludes the warping of static points, such as regions corresponding to the background or the upper torso of the body. This curation of points eases the burden on the rendering module, enhancing its capacity to learn and yielding improved results on static points. The full 3DMM deformation formula is shown in the figure 4.2. Besides, we speed up the computational process of locating the nearest vertex for each sampled point and the subsequent calculation of the distance to that

vertex by constructing a KD-tree structure [51] of the mesh at each frame, reducing the time complexity of searching from $O(N)$ to $O(logN)$.

Apart from the 3DMM-guided deformation, we train an MLP to acquire the ability to refine the deformation, thereby compensating for the inevitable errors from the coarse 3DMM-based deformation. This corrective deformation is optimized along with the rendering module in an end-to-end manner, aiming to effectively collaborate with the 3DMM Deformation toward a more accurate representation of the temporal dynamics of the scene. This MLP is designed to rectify several limitations encountered within the 3DMM deformation framework, including their failure to model regions such as the mouth interior, hair, and accessories like hairpins and eyeglasses, the sparse distribution of vertices, as well as the inevitable inaccuracies of head meshes predicted by the off-the-shelf model based on each single image. The architecture of our MLP comprises 8 layers, each composed of 128 neurons. Based on the experimental outcomes from RigNeRF, our deformation MLP conditions on the 3D position of the point, a per-frame learnable embedding and the 3DMM deformation value as input, generating a 3D corrective deformation value, which subsequently augments the 3DMM deformation. Compared with other alternative conditioning factors, such as the expression and pose parameters derived from the 3DMM model, the utilization of the 3DMM deformation value yields several advantages. It offers more explicit information, thereby enhancing the precision of the predicted deformation rectifications. Additionally, the incorporation of the positions of 3D points along with their 3DMM deformations informs the network about the expected position after undergoing deformation with the 3DMM model, hence enabling the network to make informed decisions about where and how to apply subtle adjustments.

In summary, the computation of the position for each point in the canonical space can be formulated as:

$$\mathbf{x}' = \mathbf{x} + \text{3DMMDef}(\mathbf{x}, \text{Mesh}_D) + D(\gamma_1(\mathbf{x}), \gamma_2(\text{3DMMDef}(\mathbf{x}, \text{Mesh}_D), \mathcal{D}),$$

where $\mathbf{x}$ denotes a sampled point at frame D, and 3DMMDef refers to the 3DMM deformation as elaborated earlier. D is the function to impute a corrective deformation value, which is modelled as an MLP. The terms γ_1 and γ_2 correspond to positional encoding,

serving to elevate the dimensionality from a lower space to a higher one. This augmentation is designed to ensure that the deformation accurately accounts for high-frequency regions, enabling precise corrections for closely situated yet distinctively deforming areas. The variable $\mathcal{D}$ represents the per-frame learnable embedding, an element trained with the deformation MLP and the rendering module, aiming to encapsulate the essential information required to predict the corrective deformation value but cannot be captured by the point position and 3DMM deformation.

4.1.2 Rendering Module

The deformation module serves the crucial purpose of mapping all deformed points across various frames into a consistent canonical space. In parallel, the rendering module operates on these canonical points, utilizing their spatial positions and additional contextual information to predict their colour and density. Similar to other NeRF-based approaches, our approach involves training an implicit module that captures the information to infer the colour and density for each point. A straightforward implementation of the rendering module involves a fully connected Multi-Layer Perceptron (MLP). However, this architecture shares a typical challenge with the original NeRF - extended training times due to computationally intensive operations across all layers of the MLP for each point. We consequently explore an alternative method to accelerate our training process.

Our investigation leads us to the multi-resolution hashing encoding technique to substitute the conventional positional encoding of sampled points, as the one initiated by the InstantNGP [11] model. By adopting such an encoding method, we are able to replace the deep and computationally intensive MLP with a more lightweight counterpart, without compromising the overall quality. Compared to other methods that enhance training efficiency, the multi-resolution hash offers several distinct advantages and has been successfully embedded in previous work for representing the human body and portrait [13, 49]. The multi-resolution hash encoding imparts richer and more semantically effective information by embedding the scene from diverse resolutions, capturing both coarse aspects from higher levels and finer-grained details from lower levels. Moreover, the absence of a pruning mechanism in InstantNGP proves advantageous for our purposes. In applications involving dynamic scenes, early-stage pruning of empty points

might lead to suboptimal results, as the dynamics may not have been comprehensively and exclusively learned, thereby affecting the later training and the overall performance. Additionally, our decision to adopt the multi-resolution hash encoding approach aligns with practical considerations. The existing tiny-cuda-nn (tcnn) [52] framework implements a hash grid network that aligns well with the architecture of the multi-resolution hash encoding presented in InstantNGP. By utilizing this framework, we are able to simplify our implementation and experimentation processes, facilitating our research efforts.

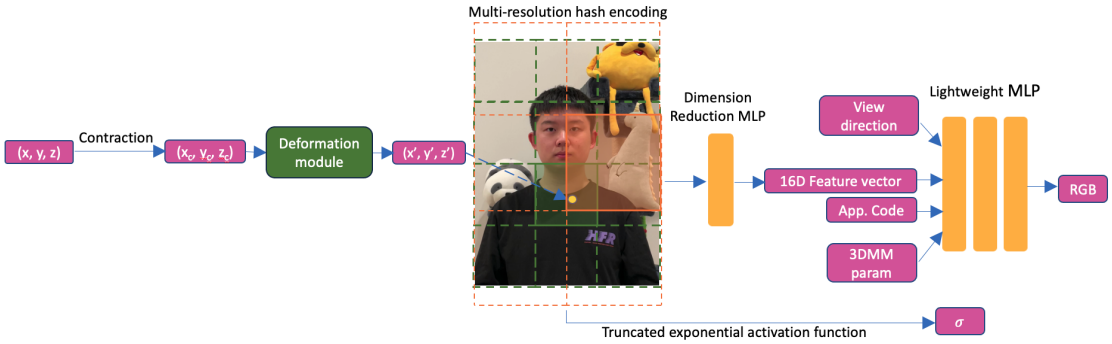


FIGURE 4.3: Our rendering module in details.

Figure 4.3 illustrates our rendering module in detail. Our rendering module comprises two parts: a multi-resolution hash encoding and a lightweight MLP. The multi-resolution hash encoding is implemented through the NetworkWithInputEncoding module within the tcnn framework, encompassing a hash grid and an MLP designed to reduce the dimensionality. This module maps a 3D input (the (x, y, z) coordinates in 3D space) into a compact 16D feature vector. For ease of the subsequent lightweight MLP, we utilize the first value of the 16D feature vector to represent the density of each point. The first value of the feature vector is processed using a truncated exponential activation function, as outlined in InstantNGP, which maps the density within the correct range of 0 to 1. The entire multi-resolution hash encoding module is optimized alongside other trainable parameters in our network, with the objective of providing the most optimal density and feature representation for each 3D point.

The role of the lightweight MLP is to generate a view and motion-dependent appearance for the portrait. Similarly to the approach in InstantNGP, except for the feature vector from the multi-resolution hash encoding, we incorporate the view direction of the ray to provide the MLP with view-dependent information. The view direction is encoded

using spherical harmonics encoding, which maps it to a higher dimension in order to capture high-frequency details. Drawing upon the approach applied in RigNeRF, we integrate motion-dependent information into the model to capture variations in the portrait appearance caused by dynamic facial poses and expressions. We augment the input of the lightweight MLP with 56D concatenated pose and expression parameters derived from DECA, as well as a per-frame learnable appearance embedding. The appearance embedding offers semantic information that cannot be captured by the pose and expression parameters. The lightweight MLP contains two layers, each with 128 neurons. It yields the RGB colour for the corresponding input point. Mathematically, the operation of the lightweight MLP can be expressed as follows:

$$c_i(\mathbf{x_i}, \mathbf{d_i}, p_{exp+pose}) = F(H(\mathbf{x_i}), \gamma(\mathbf{d_i}), p_{exp+pose}, \mathcal{A}),$$

where c_i is the inferred colour of the point, F represents the lightweight MLP, H denotes the multi-resolution hash encoding, $\mathbf{x_i}$ refers to the canonical point, $\mathbf{d_i}$ indicates the view direction, which is encoded using the spherical harmonics function γ, $p_{exp+pose}$ is the concatenated pose and expression parameters, $\mathcal{A}$ is the per-frame learnable appearance embedding. Finally, the colour for each pixel is computed based on the density and colour of all points along the ray, using the classical neural rendering approach, akin to the original NeRF framework.

One preprocessing step in applying hash encoding involves comprehending the scene boundary, a determinant that governs the dimensions and scope of the hash box. Our research objective aims to capture portraits across a diverse range of backgrounds, spanning scenarios characterized by well-defined boundaries (e.g., indoor environments with walls) as well as those without such constraints (e.g., expansive outdoor settings). Consequently, additional processing is indispensable to ensure that the model can effectively learn all points within the scene, spanning the background. Such an objective exhibits issues for conventional sampling strategies that involve defining a near and far plane and sampling points within this range. To handle unbounded scenes, contraction can transform points that could be positioned exceedingly distant from the camera. We employ the contraction technique that accommodates unbounded scenes by sampling points proportionally to their inverse distance, followed by transforming these distant points to align within a predefined bounding box.

4.1.3 Training Strategy

Throughout our experiments, we have observed that the model rapidly learns the static background but struggles with the dynamic aspects of the portrait, necessitating additional training. In light of this insight, we devised a two-stage training strategy to steer the model's focus towards the head area in the latter part of the training process. We introduced a hyperparameter denoted as B, which designates the iteration point at which the model transitions its focus towards comprehending the dynamics and appearance of the head region. B was set at 30,000 iterations in our experimentation. Within the initial B iterations, we adopted a random sampling approach akin to the vanilla NeRF when sampling rays from the images. This approach facilitates the model to simultaneously learn about the portrait and the background. Since deformation does not occur on the background rays, the rendering of static points converges rapidly, generating a clear and accurate representation of the background.

After reaching B iterations, the model shifts its focus to the head region by selectively sampling pixels biased towards this area. We utilized the mesh derived from the DECA as a reference to identify head pixels. For these specific head pixels, we increased the sampling probabilities. It's essential to highlight that a modest portion of background pixels is still incorporated, albeit with a lower probability of being sampled. This approach ensures that the background information remains relevant, preventing the head area from being learned in isolation from its background context. Then the model undergoes extended training spanning several additional thousand iterations, with the goal of refining the deformation module and enhancing the portrayal of the portrait's appearance.

4.2 Data Collection and Processing

While datasets containing sequences of portrait images captured from different viewpoints, with subjects exhibiting varied expressions and head poses, along with accompanying camera information and explicit meshes, do exist, these datasets are often compiled under controlled experimental conditions. These controlled setups often involve dedicated cameras against backgrounds that are either white or green, followed by segmentation to remove the background. However, this collection process is non-trivial

for individuals lacking expertise in the field and cannot satisfy our objective of portrait images under any background. Moreover, existing datasets that include portrait images with backgrounds primarily consist of single-view images that are curated for other tasks like portrait identification or classification.

Therefore, in our research, we determined to collect our own portrait dataset. We aim to simplify the data collection process by enabling the use of common devices like mobile phones and film portrait videos in diverse environments. To obtain the mesh for each frame, we leverage the capabilities of the off-the-shelf 3DMM DECA. It's vital to note that DECA operates within its own distinct space, separate from the world coordinate system. Hence, it's essential to establish a robust connection and transformation mechanism between the DECA coordinate system and the world coordinate system. This ensures accurate transformation of the mesh to the world coordinate system, thereby facilitating the effective functioning of the deformation module. In the subsequent sections, we will provide comprehensive insights into each facet of our data collection methodology.

4.2.1 Data Collection

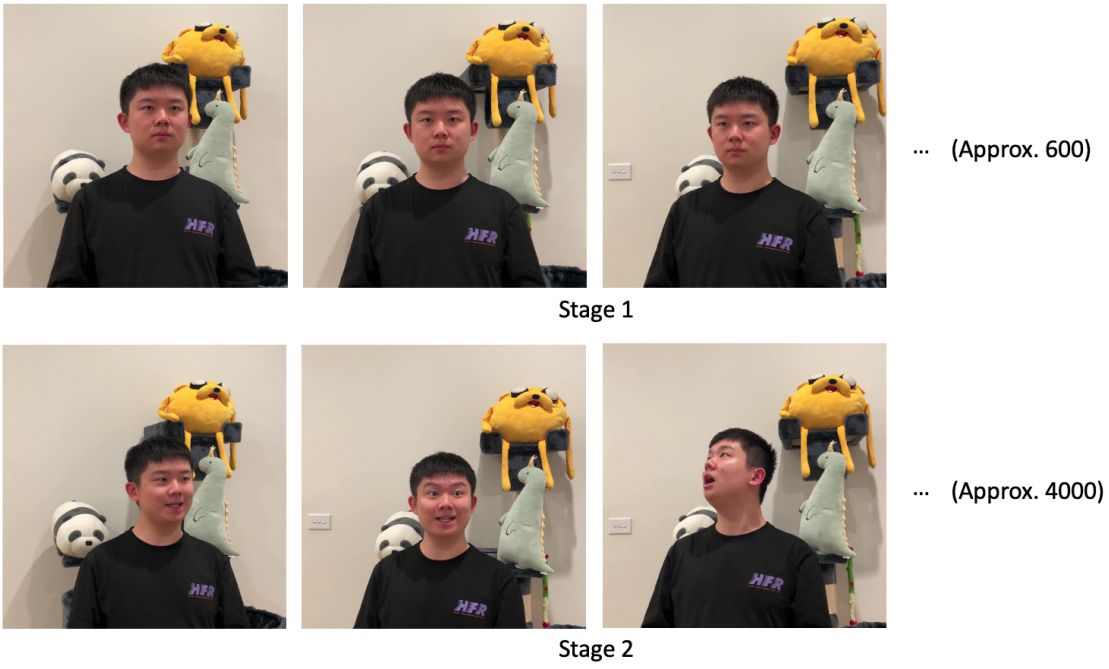


FIGURE 4.4: Example frames from Data Collection Stage 1 and Stage 2.

In this section, we will demonstrate the procedure for dataset collection, whereas we will explain the dataset processing methodologies to achieve coordinate alignment in the following sections. During the data collection process, specific instructions were provided to subjects. First of all, subjects were requested to position themselves against a static background characterized by rich textures, in contrast to backgrounds with only uniform colour or dynamic elements. Our data collection procedure is designed as a two-stage process. During the first phase of filming, subjects were instructed to maintain a stable posture, exhibiting a neutral facial expression while facing forward. The mobile phone would pan around the subject, capturing the head and a part of the upper torso of the subject from a comprehensive range of viewing angles. The first phase typically spanned a duration of approximately 40 seconds to 1 minute. Subsequently, in the latter half of the data collection process, subjects were prompted to engage in a diverse range of facial expressions, as well as imitate speech-related movements, all the while varying their head poses. The camera behaves similarly to the first stage, panning around the subject and capturing their portrait under diverse view directions. This section will last for around 2 minutes to capture a comprehensive set of facial expressions and poses, under varying viewpoints. Throughout the whole recording period, the human head should remain centred within the camera. Figure 4.4 demonstrates some example frames from stage 1 and stage 2.

4.2.2 Preliminary Data Processing

Our data collection for each scene results in two videos. Each video is then partitioned into individual frames. The presence of blurry frames has been empirically identified to adversely impact the stability and efficacy of the training process for NeRF models [5]. To address this concern, we adopt a quantifiable metric for assessing image blur, predicated on the variance of the Laplacian [53]. In essence, an image characterized by blurriness exhibits a lower variance of the Laplacian. We define a threshold criterion, which we set at a value of 30, to eliminate frames presenting a blur score below this specified threshold. This process ensures that nearly all frames are clear, mitigating potential instability issues. With the application of this selection criterion, the first video yields approximately 600 frames, while the second stage contributes around 4000 frames. Given the computational burden associated with training with high-resolution images,

and considering that our research emphasis is not on generating images of exceptionally high resolution, we have opted to implement a cropping and down-scaling strategy. Specifically, we first crop all frames to achieve a square aspect ratio by trimming equal portions from the top and bottom of each frame. All frames are then uniformly down-scaled to a resolution of 512×512 pixels. This step aligns with our objectives: our image resolution is much higher than the 256×256 pixels used in previous models such as RigNeRF, allowing for a more detailed capture of facial features and the surrounding scene. At the same time, it strikes a balance by not being excessively high, thus maintaining a manageable computational load.

We proceed to apply the DECA model, estimating the head meshes for all frames from both stages. The DECA model is invoked individually for each input image, yielding a head mesh comprising 5023 vertices. These meshes are positioned within a distinct coordinate space wherein the coordinate origin approximates the centre of head meshes. DECA also predicts intrinsic and extrinsic camera parameters associated with each image. It is imperative to clarify that these camera parameters serve the singular purpose of projecting the head mesh from its inherently head-centred space onto the image plane. It is essential to distinguish this camera parameter from the one estimated based on the multi-view information, as done in COLMAP, as the former one does not account for background information during its derivation. We project all meshes onto the image plane with the per-frame camera pose and the mesh, resulting in the alignment of the head mesh onto the image plane along the x-y axis. Due to the 2D nature of the image plane projection, the z-axis lacks any inherent restrictions and is random. Additionally, the original implementation of DECA features a pre-cropping step aimed at segmenting the head from the background. DECA will revert this cropping with the same 2D cropping transformation to ensure the meshes are projected onto the original image. However, upon inspection, we identified that this reversion process exclusively accounts for the x-y axes, neglecting the z-axis. This oversight subsequently led to inaccuracies in the scaling of the z-axis. To rectify this, we have adjusted the scaling function to encompass all dimensions (x, y, and z) concurrently. We calculate this comprehensive scaling based on the cropping scale applied to the x and y dimensions. This refined scaling strategy ensures that produced 3D meshes have correct relative scaling across all dimensions.

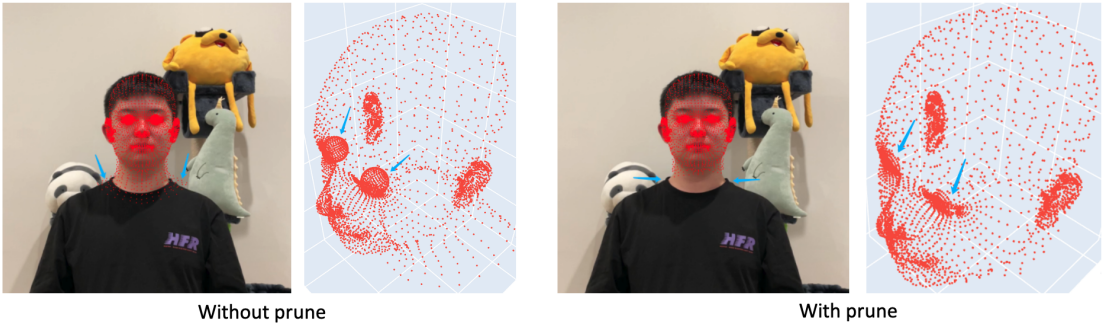


FIGURE 4.5: We prune certain vertices in the mesh to reduce complexity.

We discovered that while DECA performs outstandingly in modelling the head meshes, its outcomes in the vicinity of the neck region are less satisfactory. Furthermore, DECA's complicated vertex clustering around the eyeballs, aimed at facilitating accurate eyeball modelling for 3DMM construction, proves redundant for our purpose. These internal vertices do not significantly contribute to the final colouration with the NeRF model and may potentially disrupt deformation calculations. In light of these observations, we have undertaken a vertex pruning process (figure 4.5). We target vertices situated below the jawline and internal to the eyeball clusters. Through this curated process, we have effectively refined the head mesh, resulting in a reduced vertex count of 4069 vertices. This vertex pruning not only enhances the compatibility of the mesh with our objectives but also mitigates the potential influence of superfluous vertices on deformation calculations.

Upon this point, we have obtained frame images with their respective meshes. We proceed to register the camera parameters that were utilized during the point sampling process in our model. Following the precedent family of NeRF models, we employ COLMAP [29] to identify the camera pose in each frame. COLMAP is a structure-from-motion (SfM) and multi-view stereo (MVS) framework, able to effectively extract the camera pose of each image from a sequence of images capturing the same scene. During its calculation process, COLMAP assumes the scene to be static, using the correspondence of feature points to impute the relative camera positions. To ensure the accuracy of estimated poses and prevent feature misalignment on dynamic components (i.e. the human head), we mask out regions associated with the head, as informed by the mesh outputs obtained from DECA. As a result, only the static background components remain visible and contribute to the COLMAP registration process.

4.2.3 Coordinates Alignment

We now illustrate our methodology for addressing the coordinate alignment. Broadly, our approach involves leveraging the first stage data to acquire preliminary spatial information regarding the position of the human head within the world coordinate system. Afterwards, we transform all deformed meshes from the second stage into the world coordinate system based on the inferred first-stage mesh.

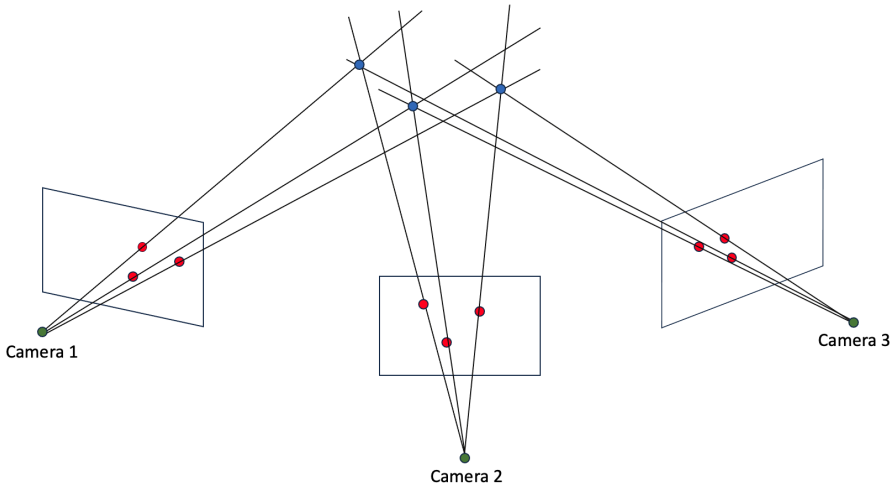


FIGURE 4.6: Triangulation visualization. By identifying the intersection point of rays originating from different camera positions, we can learn its 3D position in the world.

As we explicitly request all subjects to maintain a static posture during the first stage, an underlying assumption of our approach is that each vertex within the mesh should consistently occupy the same 3D position within the world coordinate system across all stage one frames. Armed with this assumption, we proceed to apply triangulation to infer the static mesh location. Triangulation enables us to determine the 3D position of a point based on its projections (i.e., 2D x-y coordinates on the image plane) onto multiple images captured from different viewing angles, accompanied by their respective camera poses. The fundamental principle is to identify the intersection point of rays originating from different camera positions and traversing through the corresponding 2D image points (figure 4.6). Given the importance of accurate camera calibration and feature correspondence for successful triangulation, and acknowledging potential inaccuracies introduced by our algorithmic computations (DECA and COLMAP have inevitable errors), we adopt a strategy of employing a substantial number of multi-view images. This approach is geared towards minimizing error by averaging inaccuracies and generating a sizeable set of inferred 3D points.

In our specific scenario, we utilize the x-y coordinates of each vertex on the image plane provided by DECA, coupled with the camera poses from COLMAP, to construct a 3D mesh representing the portrait head mesh in the world coordinate system. This mesh fulfils two roles. Firstly, it serves as the reference for transforming other deformed meshes into the world coordinate system. Secondly, due to its neutral facial expression and forward-facing head pose, it functions as the canonical mesh, offering a consistent baseline for deforming all other meshes into alignment.

The second stage frames serve as the training dataset. We transform their meshes by establishing a connection between the canonical mesh (world coordinate) and each individual deformed mesh (DECA coordinate). One straightforward approach entails seeking a rough similarity transformation between each deformed mesh and the canonical mesh based on all vertices and applying this transformation to align the deformed mesh with the world coordinate. However, it's important to acknowledge that due to the inherent nature of our methodology, achieving an entirely accurate transformation is challenging. This is because while the DECA meshes are subject to different deformations and variations across frames in their coordinate space, we possess a sole canonical mesh in the world coordinate with a neutral facial expression and head pose. Such a divergence leads to different transformation constraints obeyed by each corresponding point, making it tedious to register all vertices for a precise transformation.

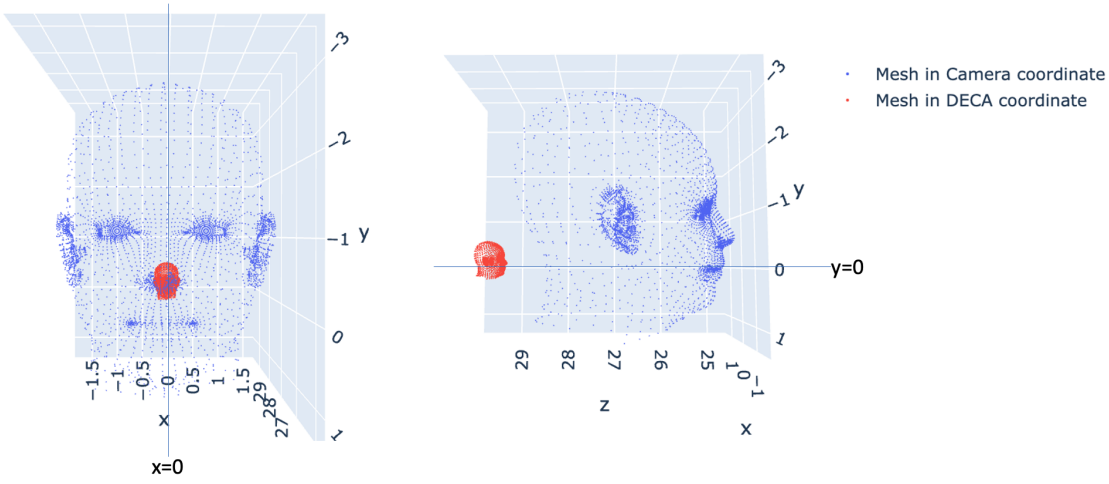


FIGURE 4.7: Demonstration of the relationship between the DECA mesh and the Camera mesh (3D).

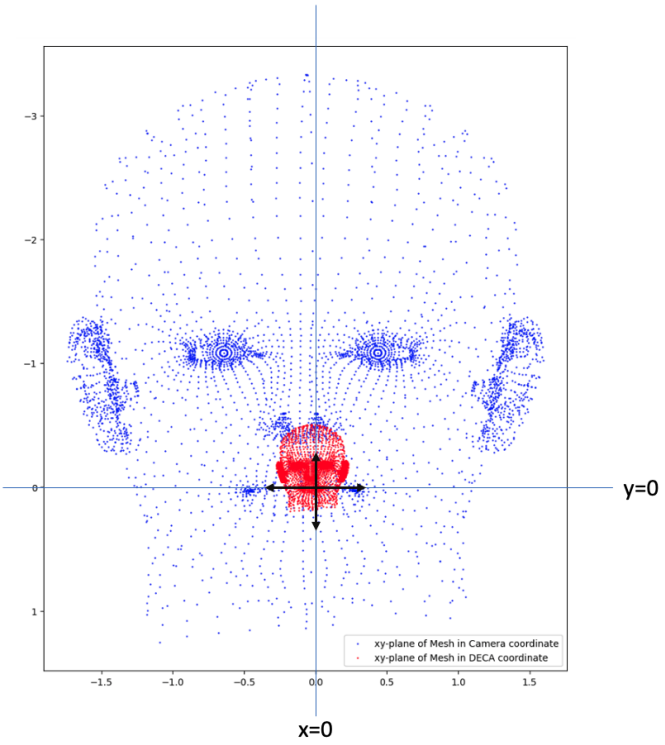


FIGURE 4.8: Demonstration of the relationship between the DECA mesh and the Camera mesh (2D, x-y axes).

The simple similarity transformation could potentially introduce excessive degrees of freedom, encompassing scenarios like disparate head rotations and incorrect mesh scaling, potentially leading to undesirable outcomes. To mitigate these complexities, we devise a more sophisticated method, capitalizing on our inherent knowledge of the DECA mesh. Notably, the x-y axes within the DECA coordinate space correspond to the image plane, albeit with an unknown z-coordinate. In addition, we can ascertain the mesh's representation in the camera coordinate system through rotation, translation, and scaling transformations indicated by the camera extrinsic. This mesh in the camera coordinate can be subsequently projected onto the image plane using solely camera intrinsic, which primarily encompasses scaling and offset adjustments for diverse image coordinate systems. We dispense with offset considerations since they can be omitted in our scenarios with the same image coordinate. Consequently, as we discern that intrinsic essentially introduces scaling from the camera coordinate mesh to the image plane, and considering that the DECA mesh is inherently situated within a space sharing the same x-y axes as the image plane (as shown in figures 4.7 and 4.8), we can compute a transformation between the DECA coordinate space and the camera coordinate space.

The transformation between the DECA coordinate and the camera coordinate specifically revolves around scaling along the x-y axes and translating the randomized z-axis to the z-axis within the camera coordinate system. For the z-axis scale, we apply a scaling factor based on the x-y scaling to ensure a consistent relative scale across all dimensions. Through this tailored approach, we deviate from a conventional similarity transformation that spans between DECA and the world coordinate. Instead, we exclusively compute an x-y scaling and z-axis translation between the DECA coordinate space and the camera coordinate space. This strategy effectively eliminates unintended rotations stemming from the transformation process, mitigates potential errors associated with translation in the x-y axes, and ensures a more robust alignment.

Despite the crafted transformation approach, we recognized its suboptimal performance in specific frames involving significant head rotations, tilting of the head and mouth opening. To address this limitation, we further optimize the transformation through the application of gradient descent. We treat the x-y scaling factor and z translation factor as trainable parameters. Our process commences by employing the previously outlined method to obtain preliminary factors, serving as initial values. These values could effectively handle most frames, although their performance degrades in frames featuring extensive deformations. We fine-tune these two parameters by defining a loss function based on the Mean Squared Error (MSE) between the x-y axes of the DECA mesh and the projected x-y axes of the transformed mesh. We treat the x-y axes of the DECA mesh as the ground truth, as they are well-aligned with facial landmarks. We leverage camera intrinsic parameters to project the transformed mesh onto the 2D image plane. Subsequently, we calculate the MSE loss between these two sets of 2D vertices. Through gradient descent, we adjust the x-y scaling factor and z-translation factor to minimize this loss. An interesting observation is that applying an identical learning rate for both parameters yields incorrect mesh positioning in the world coordinate system. To address this, we apply a higher learning rate to the scaling factor and a lower rate to the translation factor. This encourages the adjustment of the scaling factor, as the aforementioned method might encounter issues with scaling in scenarios involving mouth openings or head tilts, while being less likely to introduce errors during translation adjustments. We eventually transform the mesh from the camera coordinate to the world coordinate utilizing the camera extrinsic parameters predicted by COLMAP. Through this fine-tuning process, we achieve a significantly improved mesh position in the world

coordinate system.

4.2.4 Scene Normalization

A critical set of hyperparameters that need to be defined for each scene are the near and far planes, indicating the range within which we sample points and apply contraction. However, manually adjusting these values for each scene can be cumbersome. Therefore, we standardize all scenes to fit within a normalized space, encompassing a scale of 1 as the scene boundary, to automatically determine these two hyperparameters. This task presents several challenges. Firstly, the normalization must ensure the entire scene is enclosed within the 1-scale bounding box. Secondly, the spatial arrangement of meshes in the world coordinate system must accurately retain the relative positions of cameras and the background across frames. To achieve this, we undertake scene normalization based on a byproduct of COLMAP: the sparse scene points, which characterize the landmark points used in computing camera parameters. We calculate a transformation and a 3D scaling factor to map all points within the 1-scale bounding box. To initiate this process, we ascertain the scene centre by averaging the intersections of camera rays. Subsequently, we employ RANSAC [54] to identify the scene's ground plane based on the sparse scene points. Armed with this information, we establish a bounding box capable of including all scene points. We then compute the transformation and scaling factors to map this bounding box to a normalized box with a scale of 1. Additionally, we observed that COLMAP fails to accurately represent the true 3D positions of certain points, particularly for points close to the scene boundary because only a few images capture them. These points deviate from the main cluster of points, leading to inaccurate normalization with excessive down-scaling. To mitigate this, we exclude these outlier points (those significantly distant from the main point cluster) from the scene. To ensure the proper alignment of meshes with sampled points, we apply the same transformation and scaling matrix to both the meshes and camera positions. This normalization process alleviates the need to explicitly define near and far planes. Instead, we can uniformly sample points within a range of -1 to 1 for all scenes.

4.3 Test Data Collection

Our approach is able to create novel views of portraits with the capability of generating unseen facial expressions and head poses within the training dataset. For generating novel views, we employ a method applied in the vanilla NeRF, where the creation of test camera positions and viewpoints is based on the training camera poses. We initially compute a central camera position using all the training camera positions and determine the distance between this central position to the farthest camera position. From this, we calculate a radius that positions a series of test cameras within a sphere centred around the central camera point and is not beyond the farthest camera. We then calculate the scene centre by averaging the intersections of camera rays. We establish test camera poses that align all of them towards this determined central point. While there are alternative methods for generating novel viewpoints, such as interactive techniques involving manual viewpoint adjustments, the fundamental methodology remains the same. For the sake of experimental simplicity, we only adopt spherical test camera poses during experiments.

The synthesis of novel expressions and head poses involves modifying the head meshes of the subject. DECA provides expression and pose parameters imbued with semantic significance. By manipulating these parameters, users can generate novel expressions and head rotations. For demonstration purposes, we focus on emulating the facial expressions and head poses of other identities. This process commences by employing the encoder module of DECA to acquire the expression and pose parameters corresponding to the desired portrait for emulation. We replace the existing expression and pose parameters of the subject with these acquired parameters while keeping the identity and other attributes unaltered. This generates a mesh representative of the subject's identity but bears the expression and pose of the desired emulation. Since this mesh is situated in the DECA coordinate space, we apply the transformation approach mentioned earlier to convert the mesh into the world coordinate system. Due to the absence of ground truth, we omit the fine-tuning phase with gradient descent. We eventually utilize this transformed mesh along with the expression and pose parameters within our model, which guide the deformation network in altering the rays, thereby generating portrait images showcasing the subject with expressions and poses emulating others.

Chapter 5

Experiments

To assess the efficacy of our proposed methodology for data collection and processing, as well as the performance of our model, we conducted a series of experiments and drew comparisons with existing approaches. This chapter details the specifics of our experimental setup and the subsequent results and analyses. We start with the assessment of our data collection and processing techniques, specifically examining the accuracy of mesh transformations from the DECA coordinate system to the world coordinate system. We further analyze the improvements our fine-tuning process offers in terms of mesh coordinate transformation correctness. Following this, we describe the metrics used to measure our model's performance, with an emphasis on rendering quality and training efficiency. We quantitatively justify the performance of our model on training and validation images using these metrics with comparisons against baseline models. Furthermore, we assert models' capabilities of controlling different aspects of portraits, including generating portraits under unseen viewpoints and rendering novel portraits with expressions and head poses absent from the training data, where we emphasize a qualitative analysis due to the lack of ground truth images. Lastly, we contrast the training efficiency of our model against that of other models to understand their speed when training with a new scene from scratch.

5.1 Experimental Setup

For our experiments, we procured portrait videos of two distinct subjects adhering to the data collection protocol delineated in section 4.2.1. All recordings were captured using either an iPhone 12 Pro or iPhone 15 Pro, producing videos at 4K 60fps resolution. Each subject was filmed for approximately 4 minutes, yielding 308 stage 1 images and 3800 stage 2 images for subject 1, and 359 stage 1 images and 3440 stage 2 images for subject 2. Collected videos were subsequently processed with the method detailed in section 4.2. All models, including the variations of our developed model and the baseline counterparts, were trained on a single A100 GPU node equipped with 2 Intel(R) Xeon(R) Gold 6326 CPUs and 64GB of memory per CPU.

5.2 Data Processing Analysis

In this section, we aim to validate the correctness of our proposed data processing method. The precise mapping between the DECA coordinate and the world coordinate is vital for our model's functionality. Given that points are sampled within the world coordinate, the mesh position in this coordinate determines the accuracy of dynamic point deformation to a time-invariant canonical space. We recognize that minor discrepancies are unavoidable due to limitations inherent in the off-the-shelf 3DMM we employed since DECA offers generalized predictions based on large portrait data. Moreover, the simplicity of our data collection means it lacks multi-view information to infer the 3D details of each pose. Nevertheless, in most scenarios, the mesh should accurately approximate the position and shape of our head model in the current world coordinate, reflecting specific facial expressions and head poses without significant deviations.

We demonstrate whether our method can consistently produce meshes in the world coordinate that align with the actual head position for each frame in our two collected scenes. Given our reliance on a monocular view for each portrait pose, a definitive ground truth position is absent. To circumvent this, we adopt two qualitative assessment strategies. Firstly, we oppose head meshes in the world coordinate with sparse feature points obtained via COLMAP. This provides insights into the mesh's relative position within the scene. Next, we project meshes onto the image plane to identify which rays (i.e., pixels) intersect with mesh points. This helps ascertain if they align with the

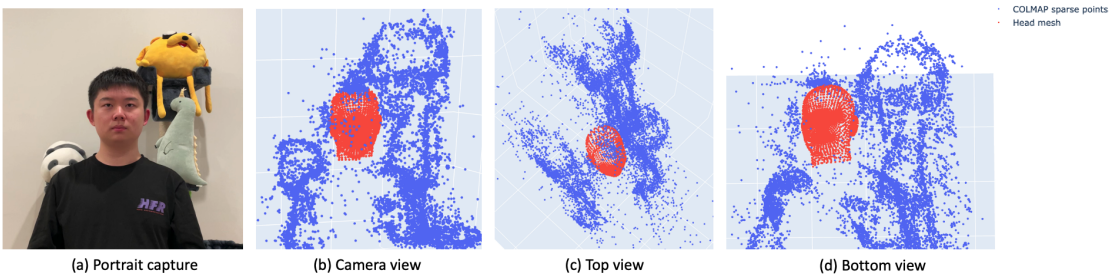


FIGURE 5.1: Example head mesh within scene points (Head pose 1 of scene 1).

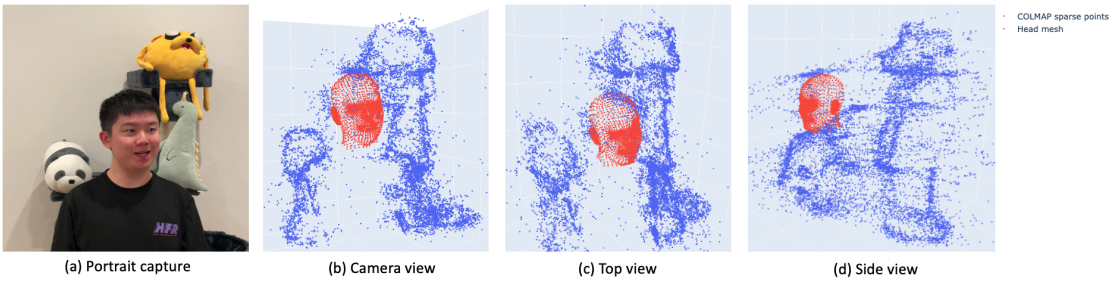


FIGURE 5.2: Example head mesh within scene points (Head pose 2 of scene 1).

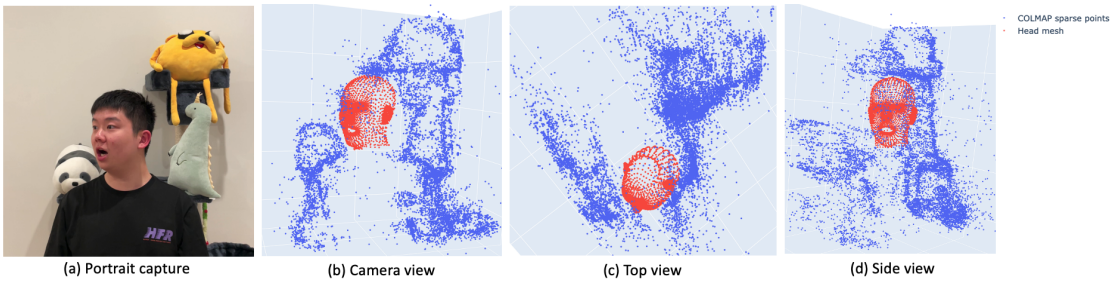


FIGURE 5.3: Example head mesh within scene points (Head pose 3 of scene 1).

corresponding 2D RGB pixels. Furthermore, we will analyze the impact of our fine-tuning process based on the DECA 2D mesh points, comparing the outcomes with and without this fine-tuning step.

Figures 5.1, 5.2 and 5.3 depict the positioning of scene points (in blue) and head mesh points (in red) within the world coordinate. We have selected the canonical mesh and two randomly selected frames with a rotated head pose as representative examples. The corresponding captured portraits for these meshes are displayed on the left for reference. From the camera angle and other perspectives, it is evident that the head mesh is appropriately positioned within the world coordinate, aligning closely with points from adjacent objects. The sparse points represent elements such as the subject's clothing and the background setting of the surrounding environment. In the first scene, the subject is seated in front of a cat climbing tree adorned with stuffed toys in a living room setting. As illustrated, the portrait mesh is accurately positioned above the cloth

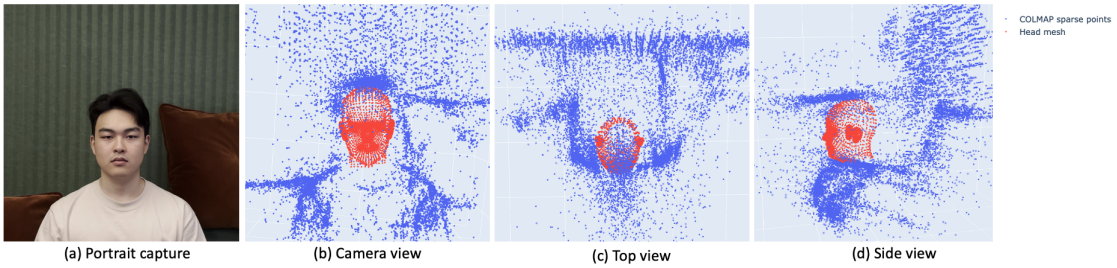


FIGURE 5.4: Example head mesh within scene points (Head pose 1 of scene 2).

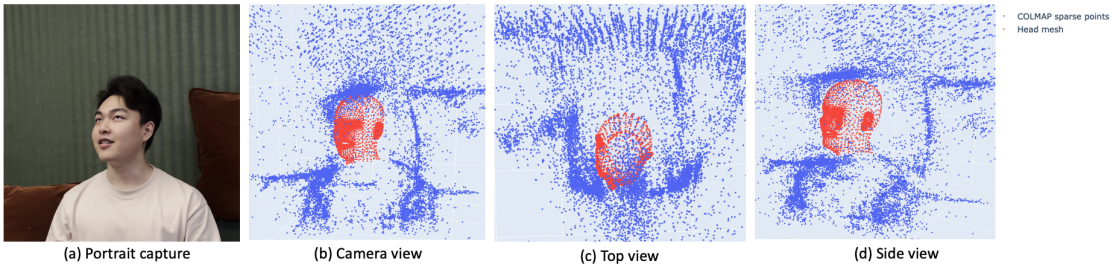


FIGURE 5.5: Example head mesh within scene points (Head pose 2 of scene 2).

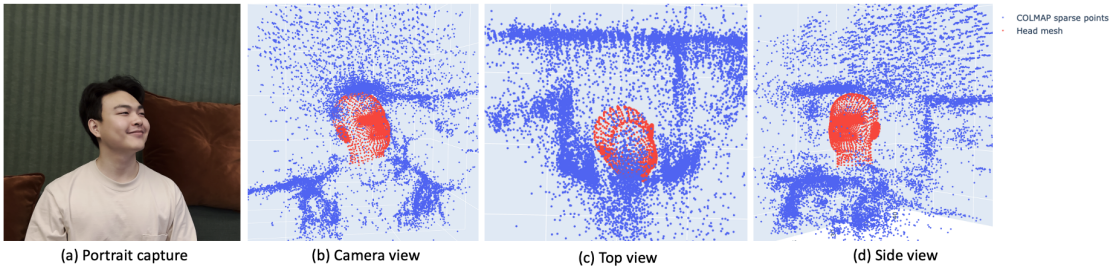


FIGURE 5.6: Example head mesh within scene points (Head pose 3 of scene 2).

points and in front of the back scene, maintaining the correct relative positioning with the cat tree and stuffed toys. In figures 5.2 and 5.3, where the head is slightly rotated to different sides and facial expression is non-neutral, the meshes accurately capture those rotations and variations in the facial region such as open mouth. The top, bottom and side views further confirm the mesh's accurate depth positioning relative to the camera. In addition, since the facial area was masked during the COLMAP training, there exists a void in the head region. The mesh's near-perfect alignment within this void further attests to the accuracy of the transformed mesh in fitting the vacant head region.

Figures 5.4, 5.5, and 5.6 illustrate the alignment of scene points with head mesh points for the second subject. These visuals underscore the adaptability of our methodology across varied settings. In this instance, the background is a relaxed office environment, with the subject seated on a sofa accompanied by cushions. Similar to the first scene, we opted for the canonical frame along with frames showcasing head rotations to provide

a comprehensive view. The results from this scene align with our observations from the first scene. The meshes are consistently and accurately positioned within the world coordinate—directly above the clothing and set against the background objects. Besides, we observe that COLMAP generates points around the facial area mask boundary. As evident in the figures, the mesh aligns seamlessly with the surrounding cluster of points corresponding to the mask edges. This further reinforces the accuracy of our methodology in positioning the mesh within the world coordinate.

While observing 3D meshes in conjunction with scene points confirms that our processed meshes approximate the human head's position within a given scene, it does not conclusively validate the precision of the mesh in mirroring the exact position of facial features of the head for each specific frame. The accurate alignment of features such as the mouth, nose, eyes, eyebrows, and ears is critical, especially when considering variations in facial expressions as these variations often arise from different combinations and movements of these facial features. To further validate the accuracy of our mesh, we project each mesh point onto the image plane using the camera's extrinsic and intrinsic parameters as predicted by COLMAP. For a camera within intrinsic parameters $K = \begin{bmatrix} \alpha_x & 0 & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix}$, where α represents the scaling factor along the x or y axis and (x_0, y_0) denotes the principal point where the optic axis intersects the image plane, and for frame i with camera extrinsic parameters including the camera rotation R (a 3×3 matrix) and translation t (a 3×1 matrix), each mesh vertex (X, Y, Z) in the world coordinate can be projected onto the image plane at point (x, y) using the equation:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = K \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & t \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}. \tag{5.1}$$

This projection allows us to observe if the projected point aligns flawlessly with the 2D facial feature points on the image. Such a projection can be perceived as the inverse operation of the point sampling process in NeRF. In NeRF, points are sampled along a ray, where each ray corresponds to a specific pixel. For a pixel associated with the mouth area, one of the sampled points along its corresponding ray should ideally match the mouth's vertex in the head mesh, factoring in the correct depth within the scene. By

FIGURE 5.7: Example head meshes projected onto the image plane (scene 1).

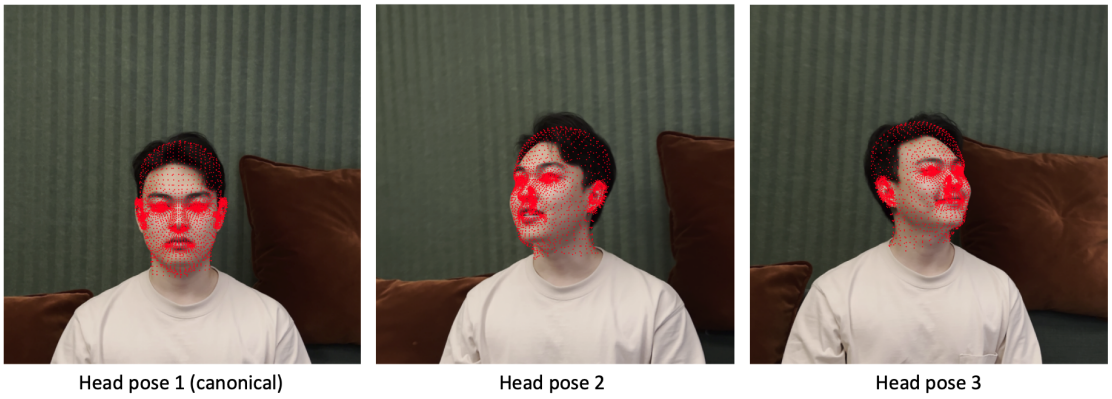


FIGURE 5.8: Example head meshes projected onto the image plane (scene 2).

projecting the mesh back onto the image and ensuring that the projected mesh aligns with the 2D pixels, we can ascertain that rays corresponding to those pixels will intersect with the corresponding mesh points.

Figure 5.7 showcases the results of projecting the head meshes from three representative frames of scene one onto the image plane, while figure 5.8 displays the results for three example frames from scene two. As evidenced by the figures, the projected mesh aligns precisely with all facial components including the eyes, nose, ears and mouth. Notably, in the mouth area, the modelling of the lips aligns seamlessly with the actual facial expressions and head poses, capturing mouth movements such as opening or compressing across different frames. The boundary of the mesh also matches the actual head's contour and size. This alignment suggests that when computing deformations, focusing solely on the facial region and a minor adjacent area, as informed by our transformed meshes, is adequate to encompass all dynamic points within the scene. Moreover, our calibrated pruning ensures that the mesh terminates appropriately at the neck region, and there are no superfluous points in areas such as around the eyeballs. These findings confirm

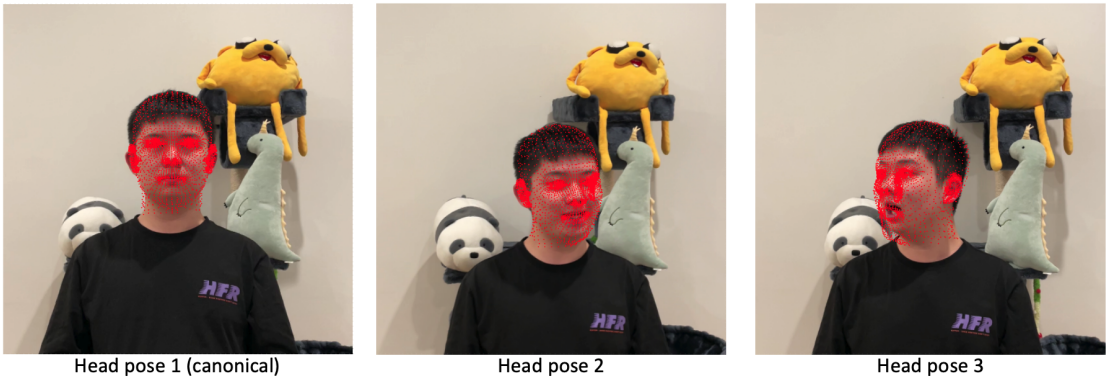


FIGURE 5.9: Example head meshes (computed without fine-tuning) projected onto the image plane (scene 1).

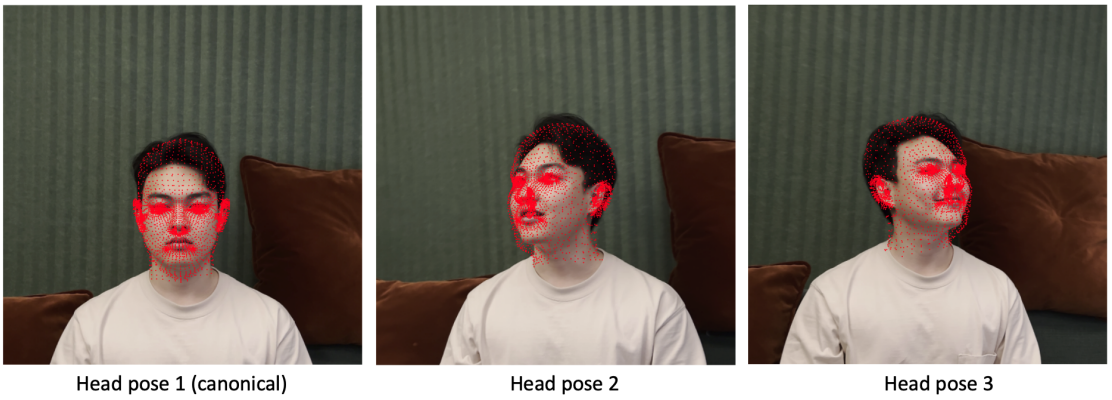


FIGURE 5.10: Example head meshes (computed without fine-tuning) projected onto the image plane (scene 2).

that rays originating from facial areas will intersect with the corresponding vertices on the head mesh within the world coordinate.

A pivotal step in our data processing methodology is fine-tuning according to the x-y axes of the meshes in the DECA coordinate. To elucidate the impact of this fine-tuning, we analyzed the projection results of meshes transformed without this process. Figures 5.9 and 5.10 display meshes of the example frames projected onto the image plane, where the mesh transformation was executed without the fine-tuning. Our findings indicate that even in the absence of this step, the majority of the transformed meshes align satisfactorily with the facial features. In the initial images of scene 1 and the first image of scene 2, where there is minimal or no head rotation, the transformation is basically accurate, and the fine-tuning process does not offer significant improvements. However, for frames with substantial deformations, such as exaggerated facial expressions like a widely opened mouth or significant head rotations, the non-fine-tuned mesh does not project perfectly onto the image plane (figure 5.11). This misalignment can be attributed

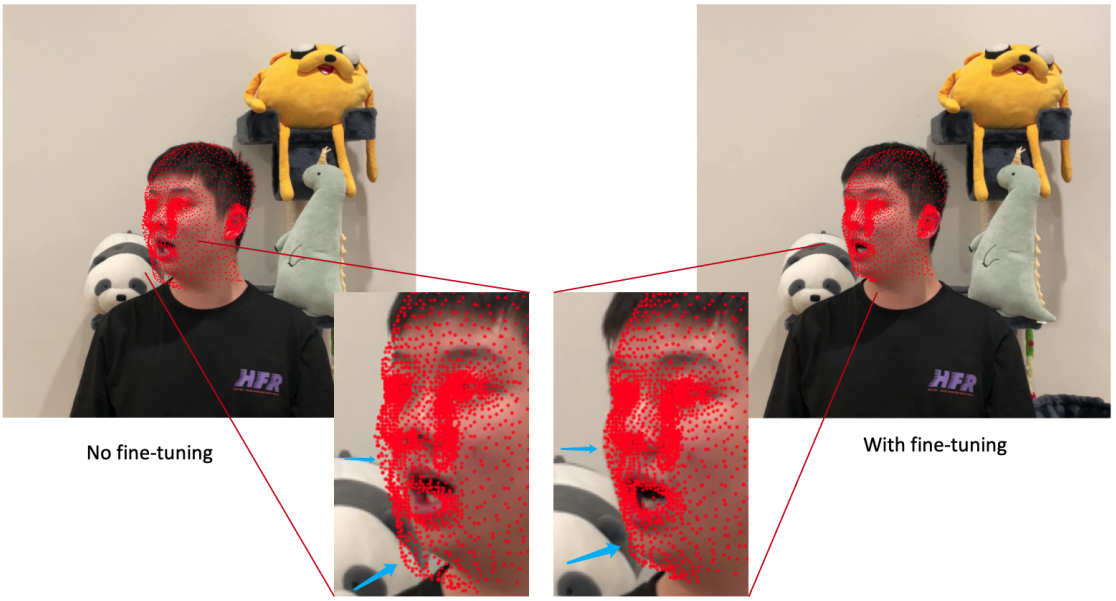


FIGURE 5.11: Comparison of projecting the head mesh transformed with and without the fine-tuning onto the image.

to our method of computing the x-y scaling factor and z-translation factor between the DECA coordinate and the camera coordinate, where we assume that the head size remains consistent and the average facial depth is invariant. However, this assumption no longer holds during significant facial expressions or head rotations. For instance, when the head is largely rotated, the centre of the head mesh may not maintain the same depth as the canonical mesh, and when the mouth is widely opened, the head length along the y-axis may vary (e.g., the cheek might be positioned lower than its location in the neutral expression). The fine-tuning process could properly address these discrepancies by adjusting the z-translation and x-y scaling to ensure that the projected mesh aligns closely with the DECA's estimation. To further substantiate the merit of the fine-tuning procedure, we present additional examples in Figure 5.12, where we showcase the projection results of meshes with and without the fine-tuning process. These comparisons further validate our assumptions and underscore the efficacy of our fine-tuning approach.

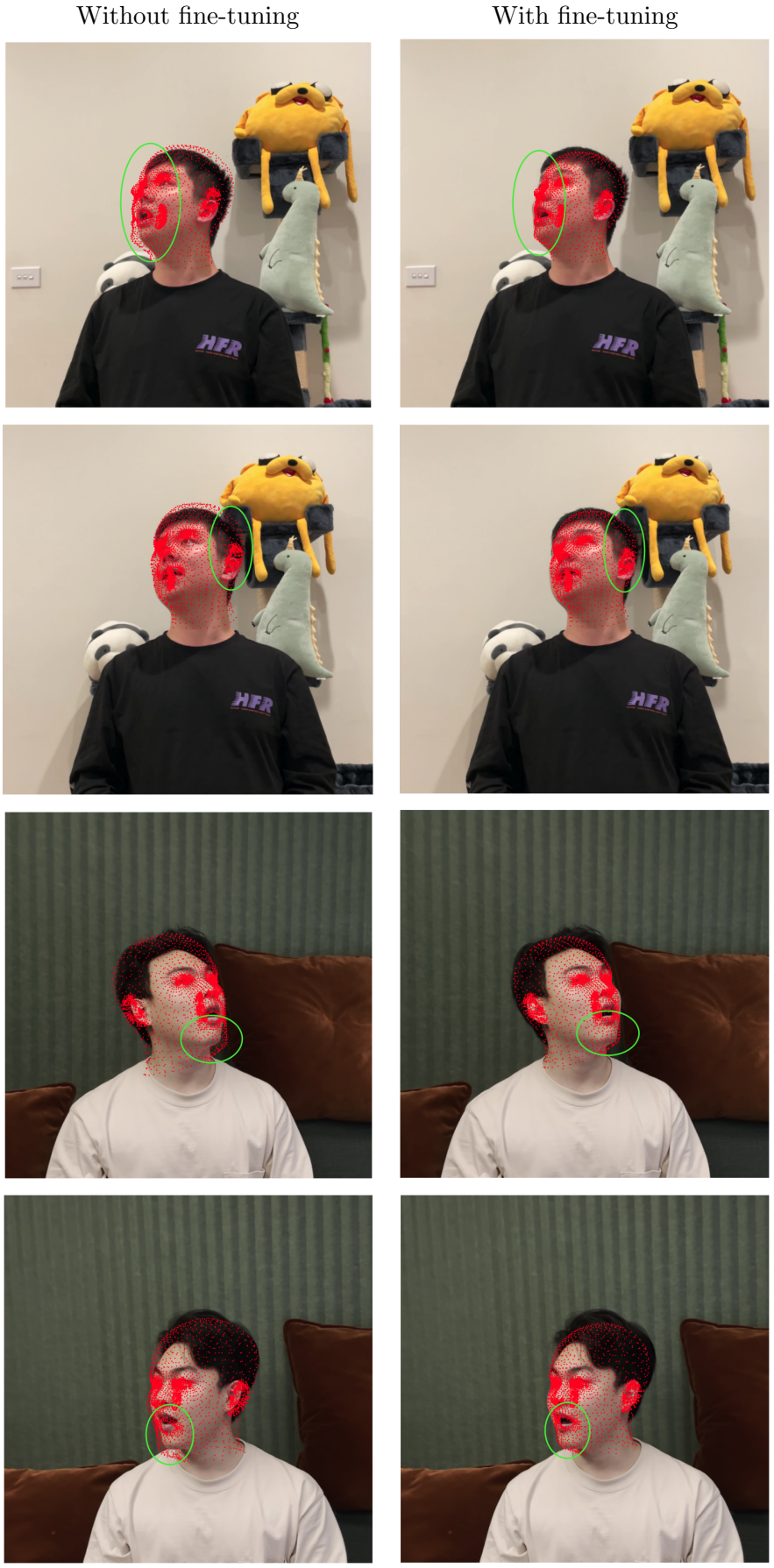


FIGURE 5.12: Comparative projections of head meshes onto the image plane. Each pair shows the same head mesh, with and without fine-tuning, to highlight the enhanced alignment through the fine-tuning process.

5.3 Evaluation matrix

We adopt a set of common metrics to comprehensively compare the performance of our models under various configurations as well as with baseline models. We assess their performance using both the training and validation datasets. From the videos we captured, 20 frames were held out and utilized as the validation dataset, ensuring that this data remained unseen during the training phase. Our evaluation of the models encompasses two primary dimensions: rendering quality and efficiency. For rendering quality, we rely on prevalent metrics for NeRF models: Mean-Square Error (MSE), Peak Signal-to-Noise Ratio (PSNR), Structural Similarity (SSIM) [55], and Learned Perceptual Image Patch Similarity (LPIPS) [56]. The MSE gauges the average squared errors between the predicted image and the ground truth, with a lower MSE signifying a closer resemblance to the original image. PSNR is derived based on MSE and evaluates the quality of image reconstruction, with higher values being preferred. SSIM moves beyond the pixel-level discrepancies captured by metrics like MSE and offers an assessment of alterations in structural information between two images. An SSIM value nearing 1 signifies a heightened degree of similarity between the images. LPIPS leverages a deep neural network to gauge perceptual differences between the inferred image and the actual image, and it assesses how a human might perceive differences rather than just numerical discrepancies. A smaller LPIPS means the inferred image is more similar to the ground truth. HyperNeRF [33] claims that PSNR and MSE are susceptible to minor deviations, while PSNR may unduly penalize clear images in favour of blurred ones. SSIM may not detect obvious artifacts. Among them, LPIPS best describes perceptual quality as human. Therefore, we place particular emphasis on LPIPS in our model evaluations.

Since all models were trained with consistent computational resources, our efficiency measurements are devoid of hardware dependency concerns. We recorded the training duration of each model in hours, with shorter durations being desirable. Given that different models might require varying numbers of training iterations to converge due to architectural differences, we additionally monitored the time duration of a single iteration and the total number of iterations that would lead to convergence for each model. Fewer number of iterations and less per iteration duration are preferred as they will consequent in a faster training process.

5.4 Models Configurations

We have selected HyperNeRF [33] and RigNeRF [8] as our baseline models, given their state-of-the-art capability in rendering dynamic 3D portraits encompassing both the human head and the surrounding environment as claimed in their papers. They also afford users the flexibility to edit specific attributes of a portrait by altering the input parameters of the model. While HyperNeRF showcases remarkable realism in rendering scenes with deformations, it lacks explicit guidance when deforming human heads, hence struggling to render portraits exhibiting unseen facial expressions and head poses. Therefore, our primary focus is comparing with RigNeRF as its original paper highlights its proficiency in rendering portraits embedded within real-world scenes from a range of perspectives. Table 5.1 summarizes the capabilities of baseline models and our model.

Models	Render novel views	Handle dynamics	Control facial expressions	Control head poses
HyperNeRF	✓	✓		
RigNeRF	✓	✓	✓	✓
3ENeRF(Ours)	✓	✓	✓	✓

TABLE 5.1: The capabilities of HyperNeRF, RigNeRF and 3ENeRF. HyperNeRF cannot explicitly control the facial expression and the head pose when rendering novel portrait images.

A crucial impediment arises as RigNeRF has not made its official code and experimental dataset publicly available. We thus have crafted our version of RigNeRF, striving to emulate its architectural design and training configurations. The RigNeRF paper lacks discussion details regarding the coordinate alignment and the acquisition methodology for head meshes. We integrated our curated data including transformed meshes to facilitate both training and evaluation. Moreover, the absence of a detailed architectural diagram and parameter processing procedure for the rendering MLP module in RigNeRF compelled us to implement our interpretative insights. We recreated RigNeRF based on the D-NeRF's official code [31]. For the MLP architecture delineated in the paper, we adhered to the original specifications: the deformation network comprises an 8-layer MLP, with each layer containing 128 neurons and a residual connection at the fourth layer. The rendering network for the prediction of a feature map and density is structured with 8 layers, each housing 256 neurons. The embedding dimensions for the appearance and the warp are set at 8.

An area of ambiguity lies in how the rendering MLP integrates various parameters like expression and head pose parameters, deformation MLP feature map, view direction and

per-frame learnable appearance embedding appearance. We argue that the expression and pose parameters from 3DMM may influence both density and colour variations under diverse poses and expressions. Similarly, features derived from the deformation network might offer insights into point deformations, potentially influencing density distribution. In our RigNeRF implementation, we input the deformation network feature, expression and pose parameters and the positional encoded 3D point into the rendering MLP, resulting in a density and a feature. Subsequently, this produced feature, combined with the view direction and appearance embedding, is processed via a smaller MLP with two layers, each containing 128 neurons, to predict the final RGB colour. In terms of encoding frequencies, RigNeRF employs 10 for point positions, 4 for ray directions, and 2 for 3DMM deformations. The model is trained with an initial learning rate of 5×10^{-4}, which is gradually reduced to 5×10^{-5} by the end of the training.

As for HyperNeRF, we strictly follow the original architecture. The model consists of a deformation module, constructed using an MLP with a depth of 6 and a width of 128, featuring a skip connection at its fifth layer. The deformation module results in a rotation and translation to transform dynamic points. The hyper-field has 64 neurons per layer and 7 layers, with a skip connection introduced at the fifth layer. The hyper-field produces a 2-dimensional output w as the high-dimension feature. The rendering network is constructed with 8 layers, each packed with 256 neurons. HyperNeRF employs 8 dimensions for latent appearance and deformation codes, with points encoded with a frequency of 6. The learning rate for HyperNeRF follows an exponential decay, transitioning from 10^{-3} to 10^{-4}.

For our model architectural configurations, we undertook a series of experiments to identify the most optimal setup. Our finalized model features a corrective deformation network composed of 8 layers, each layer populated with 128 neurons. The multi-resolution hash encoding is structured across 16 levels with differing resolutions, beginning with a base resolution of 16 grids at the coarsest level, and escalating to 4096 grids at its most fine-grained level. Each of these grids is characterized by a feature with a dimension of 2. The dimensionality reduction MLP that condenses the concatenated features from all levels into a final 16D feature vector, is composed of a single layer with 64 neurons. The incorporated lightweight MLP consists of three layers with 64 neurons in each. Both the deformation and appearance embedding have dimensions that replicate the configurations of our baseline models, set at a dimension of 8. A frequency of 8 is applied for

the positional encoding of points, where 2 is for the 3DMM deformation, and 4 is for the view direction. The model is trained with an initial learning rate of 10^{-3} decreasing to 10^{-5} by the end of the training.

5.5 Models Comparison and Analysis

5.5.1 Evaluation on Training and Validation Data

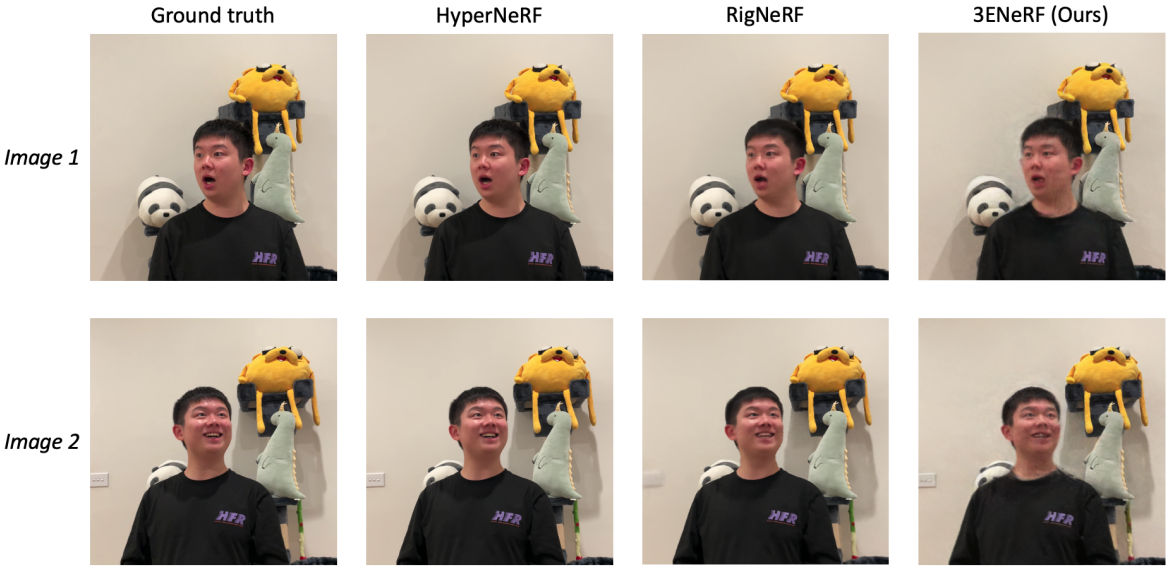


FIGURE 5.13: Comparison between HyperNeRF, RigNeRF and 3ENeRF when rendering training images (subject 1).

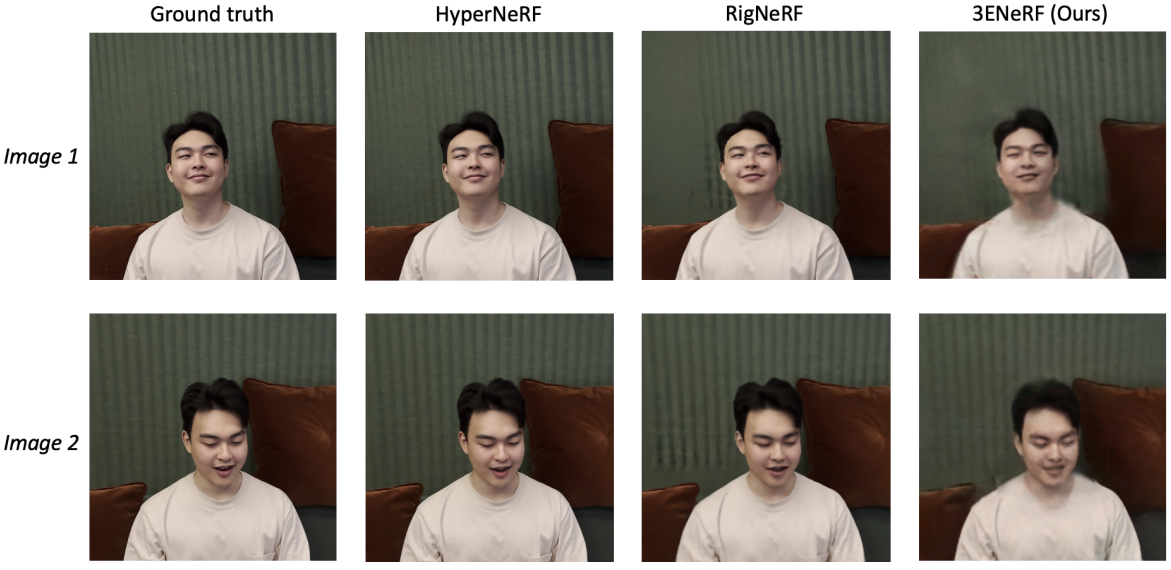


FIGURE 5.14: Comparison between HyperNeRF, RigNeRF and 3ENeRF when rendering training images (subject 2).

We trained all models on the two collected and processed data. To rigorously ascertain the quantitative performance of each model, we rendered images from both the training and validation sets, where we selected frames that highlight variations in head pose, facial expression and viewpoints. Figure 5.13 illustrates the rendering outcomes of each model for two training frames of subject 1, while Figure 5.14 depicts those for subject 2. The quantitative evaluation of these renderings is presented in Table 5.2 and Table 5.3. As shown from the figures, all of HyperNeRF, RigNeRF and our model 3ENeRF can reproduce realistic portrait images from the training dataset, even in instances of noticeable head deformations such as rotations or mouth openings. The rendered images are not only perceptually congruent with the original but also reconstruct the human head with discernible facial features while simultaneously preserving the integrity of the entire 3D scene.

Models	MSE↓	PSNR↑	SSIM↑	LPIPS↓
HyperNeRF	4.7965	41.3216	0.9834	0.0123
RigNeRF	8.7921	38.6899	0.9658	0.0224
3ENeRF(Ours)	16.3930	35.9842	0.9342	0.0602

TABLE 5.2: Quantitative comparison between HyperNeRF, RigNeRF and 3ENeRF when rendering training images (subject 1).

Models	MSE↓	PSNR↑	SSIM↑	LPIPS↓
HyperNeRF	3.9614	42.1526	0.9775	0.0173
RigNeRF	14.1262	36.6305	0.9369	0.0449
3ENeRF(Ours)	25.7329	34.0259	0.8719	0.1052

TABLE 5.3: Quantitative comparison between HyperNeRF, RigNeRF and 3ENeRF when rendering training images (subject 2).

Among the models, HyperNeRF demonstrated the most superior results, evident by its leading quantitative metrics as presented in Table 5.2 and Table 5.3. This underscores HyperNeRF's astonishing capability in precise point deformations, facilitated by its deform network, as well as its proficiency at capturing appearances across frames through its hyper-field. RigNeRF, though slightly lagging behind HyperNeRF, still presented commendable scores, indicating our head meshes are accurate in the world coordinate to facilitate a correct 3DMM-guided deformation. Conversely, our model's performance was found to be slightly worse than others in metrics like MSE and LIPIPS when benchmarked against the training images. While, at a glance, its rendered images appear largely identical to the original, a deeper examination reveals notable discrepancies and artefacts in high-frequency areas such as the corners of the eyes and mouth interiors. For

subject 1, despite our model's capability of handling dynamic head movements and facial transformations, some fine-grained details were suboptimal. Issues such as missing teeth, inaccurate tongue representation, and blurry regions near face edges were observed. Other facial elements such as the skin surrounding the cheek, jaw and forehead also exhibit an excessive smoothness. The shortcomings were even more perceivable for subject 2, with blurring being particularly conspicuous around the eyes. The rendered images give the impression of closed eyes when the corresponding frames are open. The contour of the head lacks sharpness, resulting in diminished fidelity.

We postulate that these challenges stem from the inherent characteristics of the multi-resolution hash encoding employed in our model. Despite its granularity, it remains outperformed by positional encoding in high-frequency areas due to its inability to distinctly separate proximate points. This inherent limitation, acknowledged even in the original paper, results in blurring due to the averaging of nearby points. Additionally, for portrait rendering, the appearance and geometry of the human head are influenced not solely by the 3D location of a point. Factors such as facial expressions, head poses, and the relationship between the facing and viewing directions also play pivotal roles. While models like RigNeRF and HyperNeRF address these complexities with expansive and dense MLPs comprising more than eight layers, a lightweight MLP might struggle with simultaneously processing this vast range of information. This can lead to blurred results, as the model may resort to averaging appearances across all frames to minimize loss over the entire dataset. Furthermore, the 3DMM's constraints in capturing intricate deformations, especially in areas like the mouth interior or along facial edges, compound the issue. The sparse nature and inevitable modelling discrepancies of 3DMM might account for the subpar performance of both RigNeRF and our model in comparison to HyperNeRF.

We similarly chose two images from the validation dataset, ensuring they were not exposed to models during training. We render these two validation images with all models. For determining the deformation and appearance embedding, we employed an averaged embedding approach. Specifically, the embedding was imputed as a mid-point value between adjacent training frames. We used the embeddings from their nearest neighbours for frames that are not in the middle of two training frames (i.e. the first or last frame in the sequence of frames).

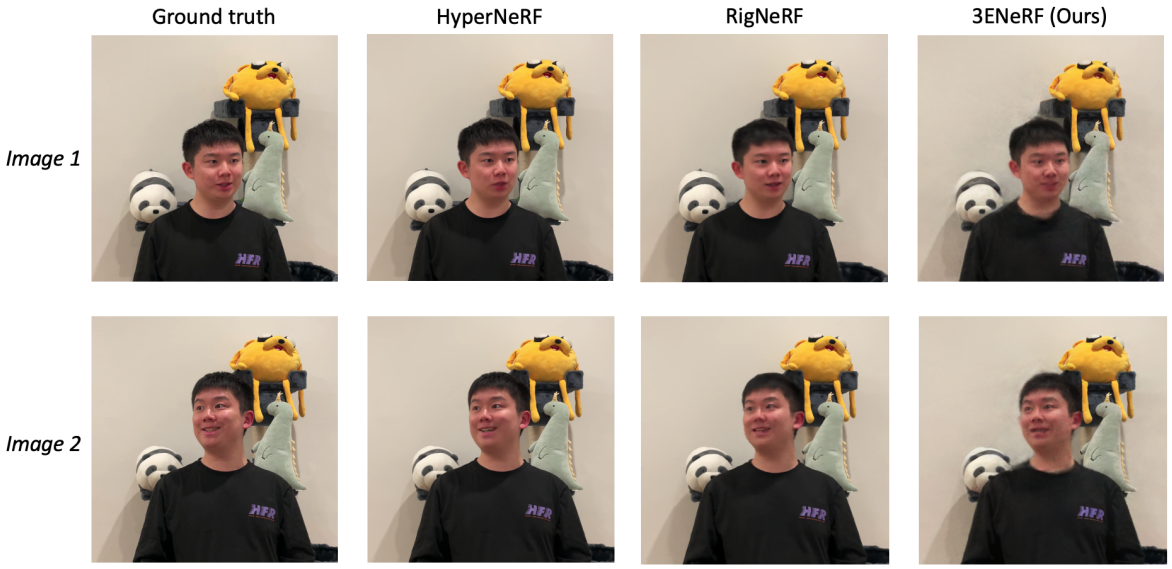


FIGURE 5.15: Comparison between HyperNeRF, RigNeRF and 3ENeRF when rendering validation images (subject 1).

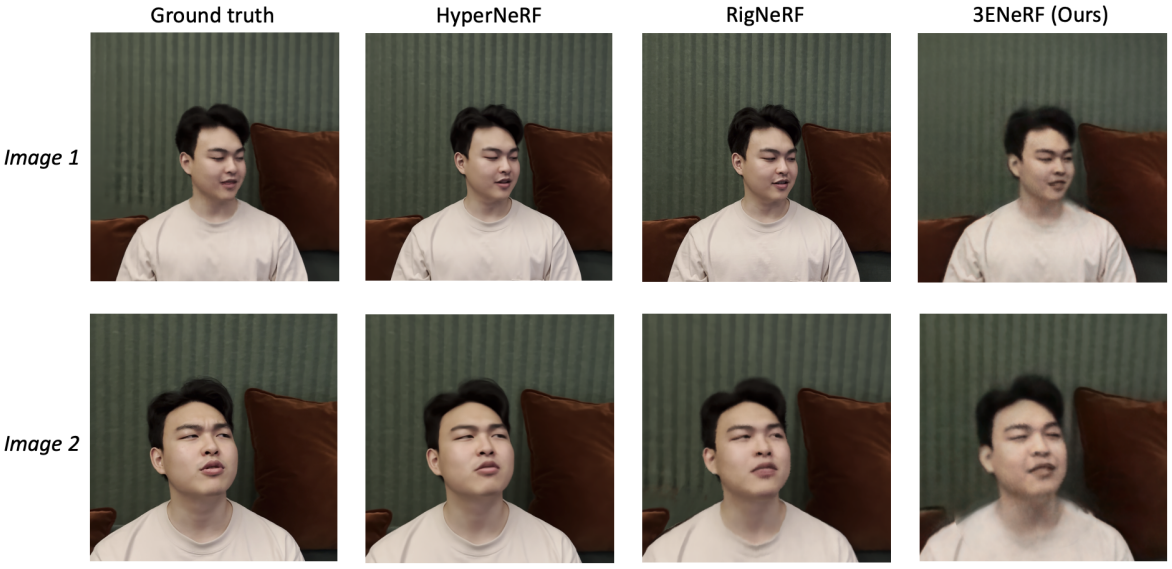


FIGURE 5.16: Comparison between HyperNeRF, RigNeRF and 3ENeRF when rendering validation images (subject 2).

Figures 5.15 and 5.16 illustrate that all three models retain consistent performance while rendering unseen images, akin to their outcomes on the training dataset. This attests to their adaptability to novel frames, indicating that our model's capability to render unseen frames is on par with its predecessors. Tables 5.4 and 5.5 suggest the quantitative evaluations of each model's performance on the validation dataset. From the tables, HyperNeRF continues to outshine the others, followed by RigNeRF and our 3ENeRF. This proves HyperNeRF's proficiency in generalizing for frames proximate to the training frames, given that our validation frames are interspersed among the training

frames. However, a slight degradation in performance, as indicated by marginally inferior metrics, suggests potential overfitting, which could be attributed to its complicated architecture and relatively abundant parameters. Conversely, both RigNeRF and our model demonstrate near-identical performance on the validation frames as they did on the training dataset, implying that their performance remains consistent across unobserved frames. Yet, their results still fall behind those of HyperNeRF. For our model, blurring remains existence in areas like facial features, face contour, and certain skin regions. A notable observation is the near-consistent closed-eye appearance for subject 2. This might stem from the preponderance of training frames for subject 2 featuring closed or nearly closed eyes. Averaging across these frames inadvertently biases the model towards rendering closed eyes. This behaviour suggests that the combination of hash-encoding with a lightweight MLP may not match the efficacy of other methodologies while processing subtle elements.

Models	MSE↓	PSNR↑	SSIM↑	LPIPS↓
HyperNeRF	5.8100	40.4890	0.9761	0.0189
RigNeRF	10.4674	37.9324	0.9592	0.0258
3ENeRF(Ours)	16.8537	35.8638	0.9354	0.0710

TABLE 5.4: Quantitative comparison between HyperNeRF, RigNeRF and 3ENeRF when rendering validation images (subject 1).

Models	MSE↓	PSNR↑	SSIM↑	LPIPS↓
HyperNeRF	8.0344	39.0813	0.9592	0.0282
RigNeRF	13.3855	36.8644	0.9401	0.0567
3ENeRF(Ours)	30.7360	33.2543	0.8629	0.0883

TABLE 5.5: Quantitative comparison between HyperNeRF, RigNeRF and 3ENeRF when rendering validation images (subject 2).

While the performance of 3DMM-based models (including ours) may not match that of HyperNeRF in rendering training and validation images, their accuracy in rendering dynamic head motion is still commendable. This suggests that warping points via leveraging the prior information from head meshes in the world coordinate is generally correct and effective.

5.5.2 Evaluation on Novel View Synthesis

We had quantitatively examined the performance of novel view synthesis of HyperNeRF, RigNeRF and our model leveraging the validation dataset which comprises viewpoints

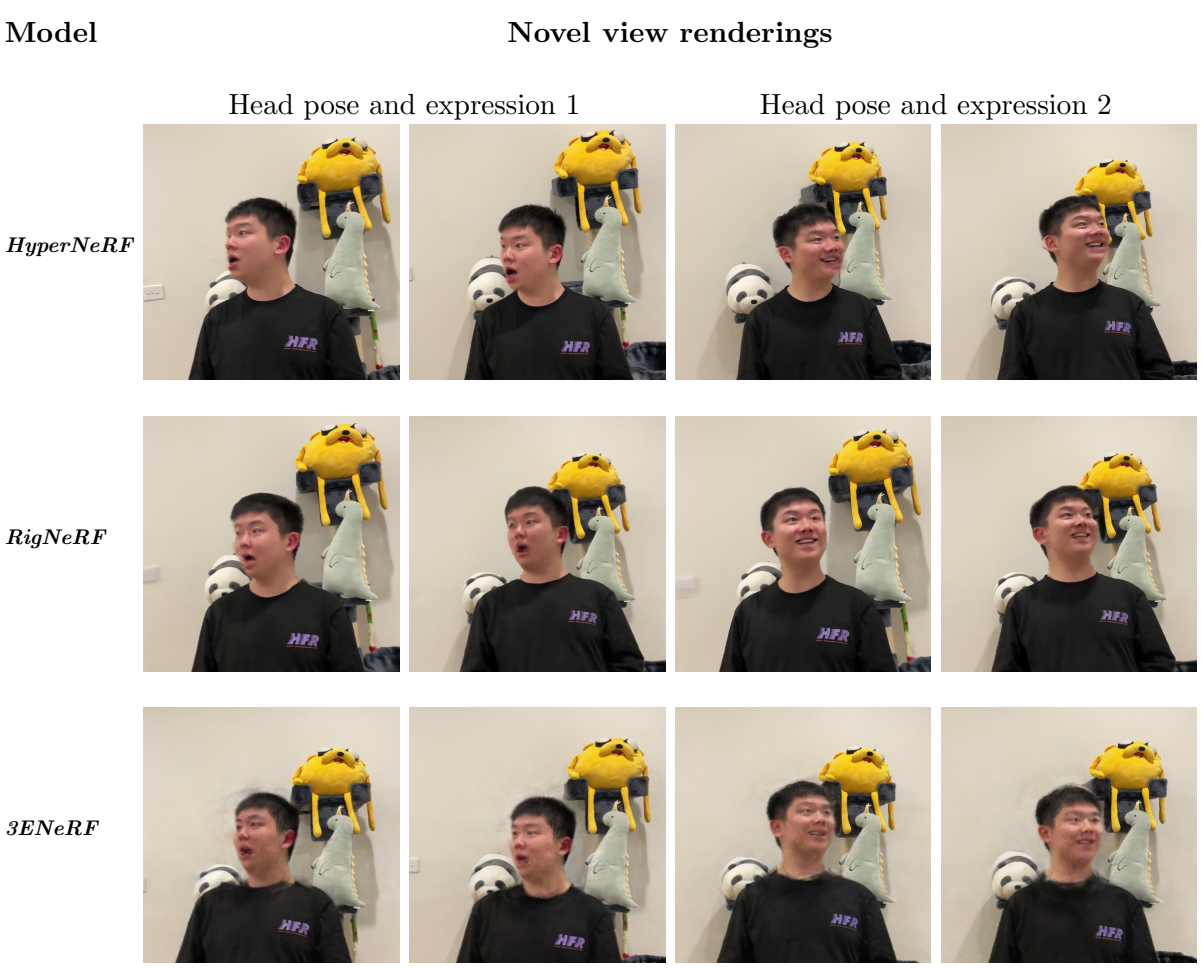


FIGURE 5.17: Qualitative comparison between HyperNeRF, RigNeRF and 3ENeRF when rendering portrait images under novel viewpoints (subject 1).

that were not part of the training set in the preceding section. However, these validation frames were spread within sequences of training images. Their viewpoints essentially lie between two training frames, making them only marginally divergent from the training perspectives. This subtle positioning might not fully reflect the model's effectiveness in rendering from genuinely novel viewpoints. To address this gap, we curated a testing trajectory based on the training viewpoints and directions but intentionally included perspectives distinct from the training set as detailed in section 4.3. Figure 5.17 presents renderings of subject 1 head poses and expressions under some example novel viewpoints across all models, with Figure 5.18 doing the same for subject 2. For reference, we used the same facial expressions and head poses that were demonstrated when rendering training images, with the original viewpoints depicted in Figures 5.13 and 5.14.

As illustrated, all models capably rendered portraits under these novel views, generating images where both the human head and the background scene vary in a cohesive

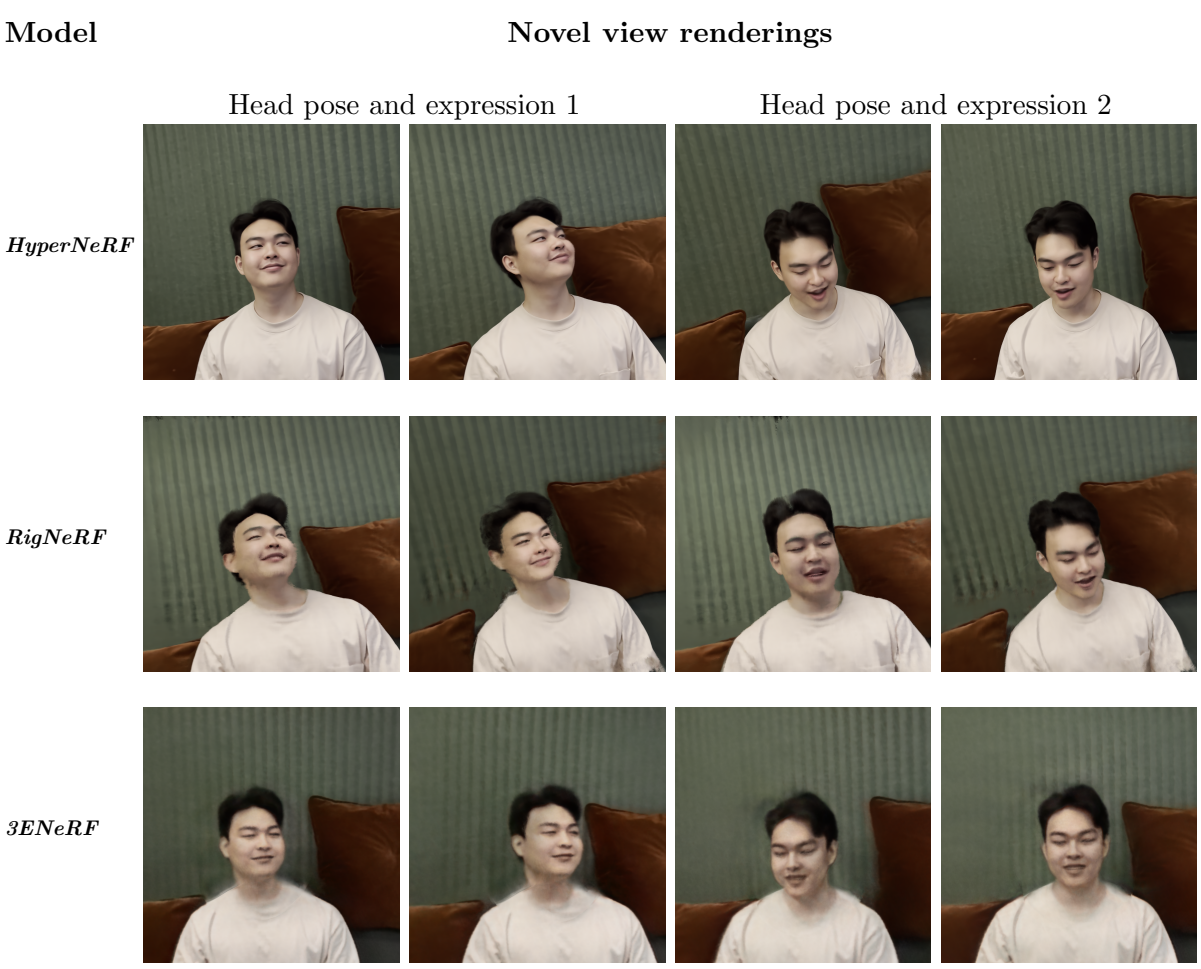


FIGURE 5.18: Qualitative comparison between HyperNeRF, RigNeRF and 3ENeRF when rendering portrait images under novel viewpoints (subject 2).

manner. HyperNeRF consistently delivers the best results for subject 2, maintaining the integrity of head structures and facial expressions. However, a closer inspection of Figure 5.17 reveals some anomalies. When the viewpoints deviate considerably from the originals, HyperNeRF appears to introduce distortions that compromise the authentic head structure. This observation becomes even more perceptible in Figure 5.19 which offers a magnified view of HyperNeRF's renderings in the head regions. From this figure, the clarity, sharpness and photo-realism that HyperNeRF previously demonstrated seem to wane. Facial features, especially around the eyes and mouth boundaries, appear blurred, with distortions making facial elements like the nose and cheek appear broader than the real underlying face. As shown in the second instance in this figure, there are also scenarios where the lips are not positioned correctly. Such discrepancies might suggest that HyperNeRF is overfitting to the training images, leading to misalignment between the inferred scene geometry and the true geometry when confronted with novel

FIGURE 5.19: Close-up of the facial region of HyperNeRF novel viewpoints rendering.

viewpoints.

On the other hand, 3DMM-based models, RigNeRF and our 3ENeRF, seem to retain the original head shape and boundaries even under substantially different viewpoints as shown in the figures, suggesting that they are less likely to be overfitting. This resilience can be attributed to the guidance provided by the head meshes generated by pretrained 3DMM. With such a geometric prior in place, these models do not learn the geometry of the underlying head mesh in isolation but rather build upon an established foundation. In contrast, purely MLP-based methods like HyperNeRF have the risk of overfitting the training images, subsequently leading to inaccurate head shapes. This underlines the merits of leveraging prior information from head meshes in the world coordinate, especially when rendering from novel viewpoints. In terms of the rendering quality of the portrait images, RigNeRF and our model behave similarly to their performance on the training and validation datasets. In contrast, RigNeRF's output is slightly sharper, while our model tends to exhibit a certain degree of blurring. Overall, however, their ability to render consistently across different viewpoints is evident and appreciated.

5.5.3 Evaluation on Novel Expression and Pose Synthesis

A primary objective of our model is the capability to render unseen portraits while granting users the freedom to modify facial expressions and head poses. HyperNeRF

does not possess this capability as it relies on a deformation embedding for each frame to render dynamic expressions and poses. It handles dynamics implicitly which means that there is no straightforward method to generate new deformation embeddings that would produce a desired expression or pose. Alternatively, both RigNeRF and 3ENeRF are empowered with this capability ascribed to their 3DMM-guided deformation approach. Theoretically, users can produce portraits with arbitrary facial expressions or head poses by generating meshes that correspond to the expected facial expressions and head poses as elaborated in section 4.3. We demonstrate such capability via face reenactment, where given a novel facial expression and head pose of another person, we reproduce such a face while preserving the identity of the original subject. We photographed a third subject, ensuring the capture of previously unseen expressions and head positions.

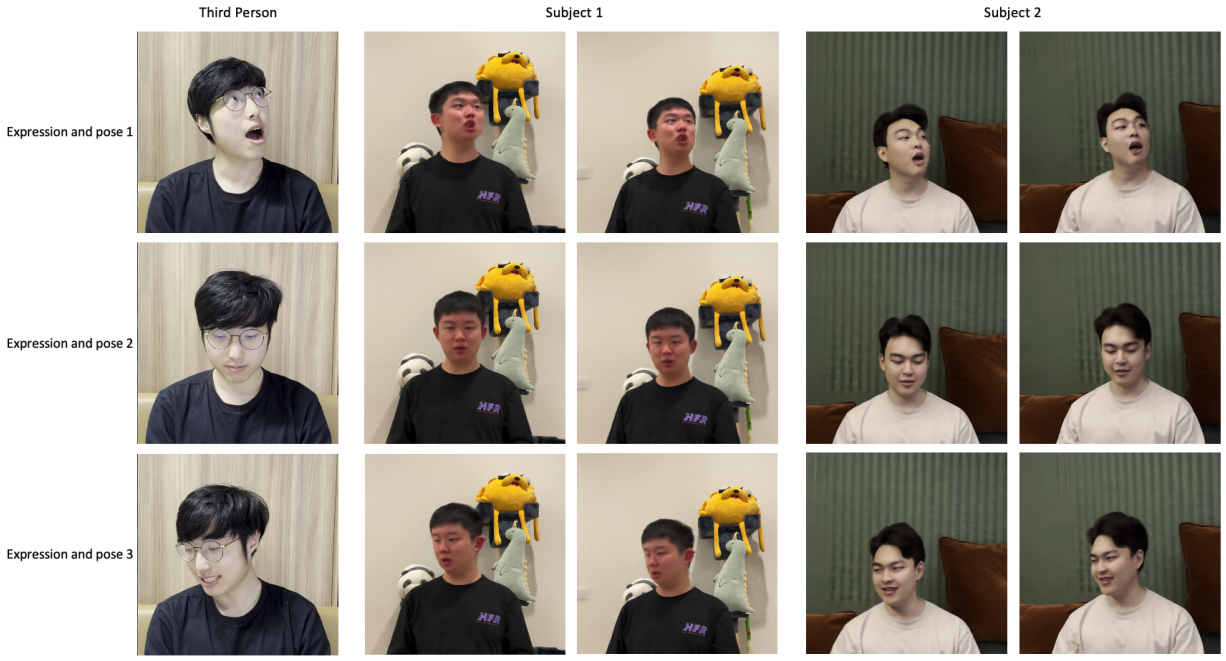


FIGURE 5.20: Novel head pose and facial expression reenactment by RigNeRF.

The reenactment outcomes of RigNeRF and 3ENeRF are illustrated in figures 5.20 and 5.21 respectively. In these demonstrations, portraits are rendered with the expressions and poses of a third individual from two perspectives: a frontal, central viewpoint and a random novel viewpoint from the testing trajectory. As evident from the figures, both models proficiently render portraits that emulate another individual's expressions and poses, while maintaining fidelity comparable to their performance on training, validation, and novel view rendering. Consistent with previous experiments, RigNeRF

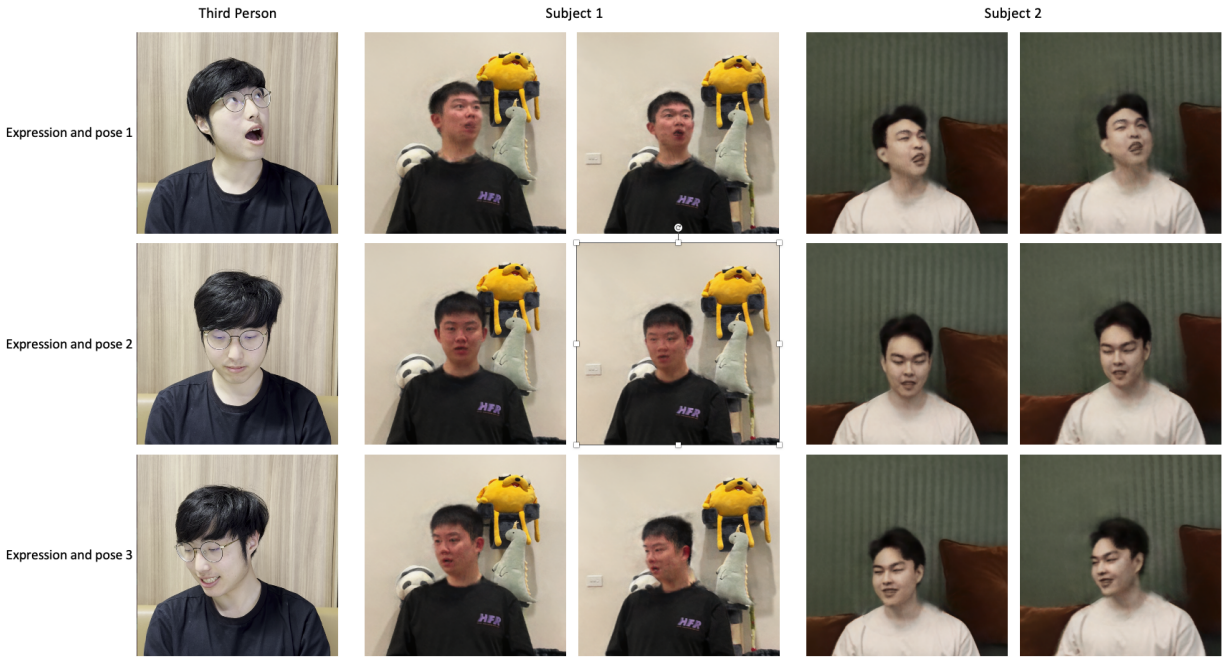


FIGURE 5.21: Novel head pose and facial expression reenactment by 3ENeRF.

delivered better outcomes, capturing high-frequency details with less blurriness compared to 3ENeRF. However, while certain facial variations such as mouth opening and head rotations are rendered in congruence with the third individual, the reenactment is not always perfectly aligned. For instance, in the second head expression and pose, the mouth of the third person is firmly closed, yet the rendering from both models for both subjects displays a slight mouth opening. Such discrepancies might arise from the 3DMM's occasional inaccuracies, which could predict a mesh that suggests a slight mouth gap and lead to the observed misalignment. Similarly, in the third set of poses, the individual to be reenacted visibly opens their mouth and reveals his teeth. However, the representation for the first subject omits the teeth, and for the second subject, the teeth appear blurred. This shortcoming can be accountable to the 3DMM's lack of detailed modelling for certain elements such as teeth.

In spite of these constraints stemming from the limitations in head mesh modelling and the occasional mispredictions by the 3DMM, the capability to reenact another individual's unseen expressions and head movements is appreciable. Both RigNeRF and 3ENeRF showcase promising results in generating novel head expressions and poses, while simultaneously granting users the flexibility to alter viewpoints.

5.5.4 Training and Rendering Efficiency Analysis

Given that NeRF-based models are required to be trained from scratch for each individual scene, the duration to thoroughly learn scene information is critical. High training efficiency is essential, especially if one envisions its application in practical settings. For our experiments, each model was trained until its training loss was no longer massively reduced. As the number of iterations required for convergence can be influenced by the specific architecture of the model, we visualize the training progression of models using their learning curves.

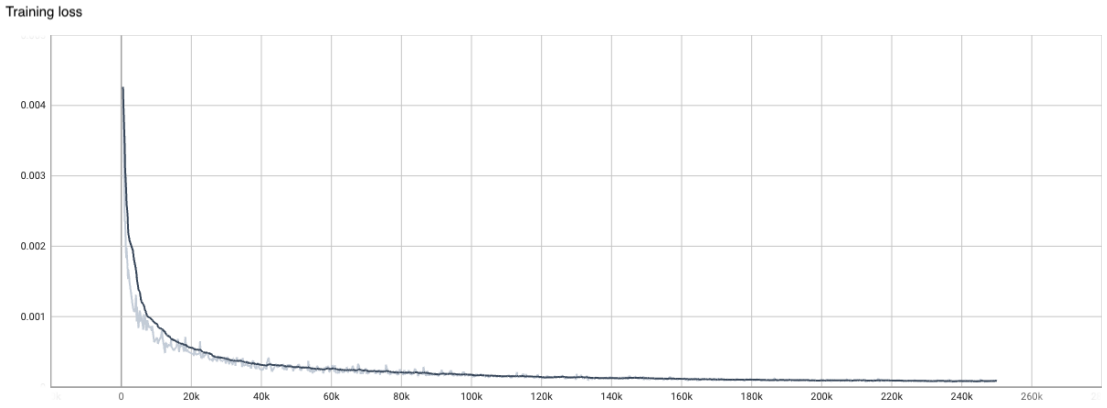


FIGURE 5.22: Learning Curve of HyperNeRF (Subject 1).

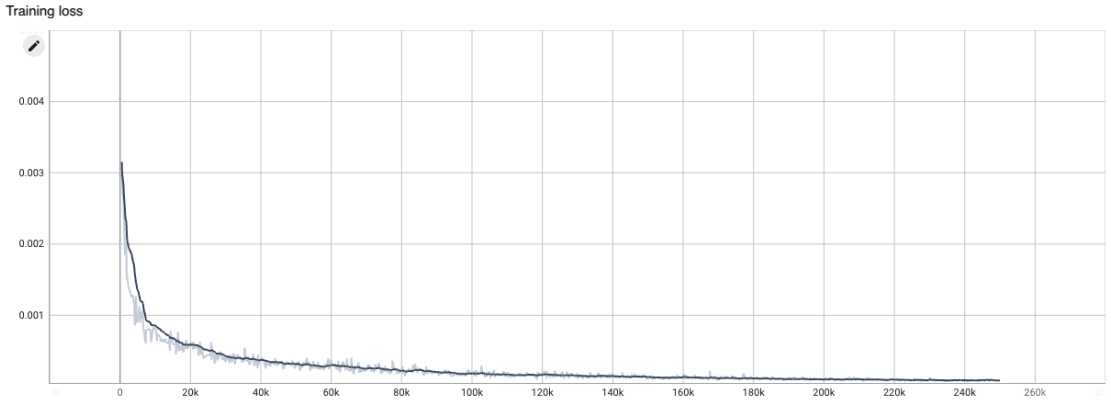


FIGURE 5.23: Learning Curve of HyperNeRF (Subject 2).

The learning curves for HyperNeRF are depicted in Figures 5.22 and 5.23. Both curves demonstrate a similar trajectory: there is a precipitous decline in loss to approximately 0.0008 within the initial 15,000 iterations. Subsequently, the rate of this decline moderates but persists until about 140,000 iterations. Beyond this point, the decline is more gradual, eventually stabilizing around 200,000 iterations with a concluding loss value of

approximately 0.0002. This trend suggests that HyperNeRF can understand the overarching geometry and appearance of the scene relatively swiftly within the initial 20,000 iterations. However, refining this understanding and converging to a minimal loss demands an extended training duration, with almost 180,000 additional iterations. The extremely low final loss underscores its aptitude to superbly fit the scene, which also hints at a potential to overfit the training images.

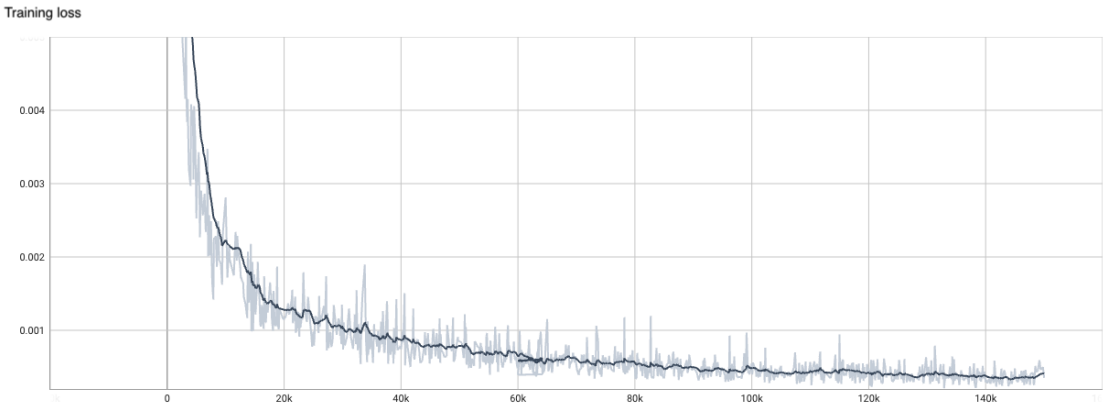


FIGURE 5.24: Learning Curve of RigNeRF (Subject 1).

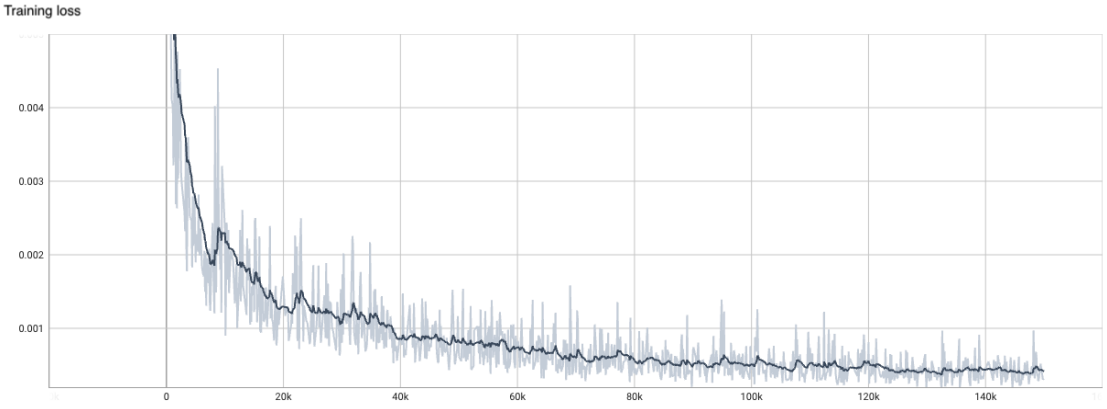


FIGURE 5.25: Learning Curve of RigNeRF (Subject 2).

Meanwhile, the learning curves for RigNeRF, as presented in Figures 5.24 and 5.25, reveal a slightly divergent pattern for the two subjects. There is an uptick in the learning curve for subject 2 around the 10,000 iterations, possibly due to the sampling of edge viewpoints which leads to a transient surge in the loss. Nevertheless, the general trajectory remains similar between the two subjects and akin to that of HyperNeRF, where a pronounced reduction in loss to roughly 0.001 is observed in the initial 30,000 iterations. This decrease then moderates, continuing until around the 150,000 iterations where it levels off, settling at a loss of about 0.0004. Compared to HyperNeRF, RigNeRF exhibits a more gradual rate of loss reduction and converges to a marginally higher final

loss, which aligns with the prior demonstrations of HyperNeRF's superior performance in rendering training images. However, RigNeRF converges to a stable loss faster, further advising it might be more resilient against overfitting to the training data.

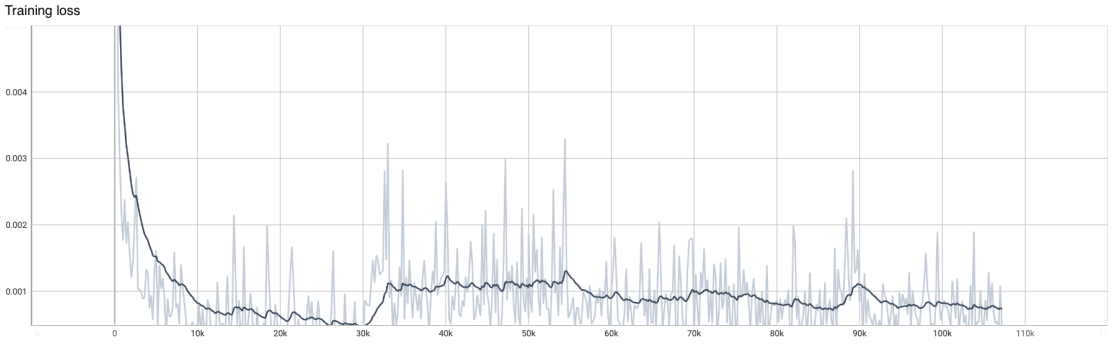


FIGURE 5.26: Learning Curve of 3ENeRF (Subject 1).

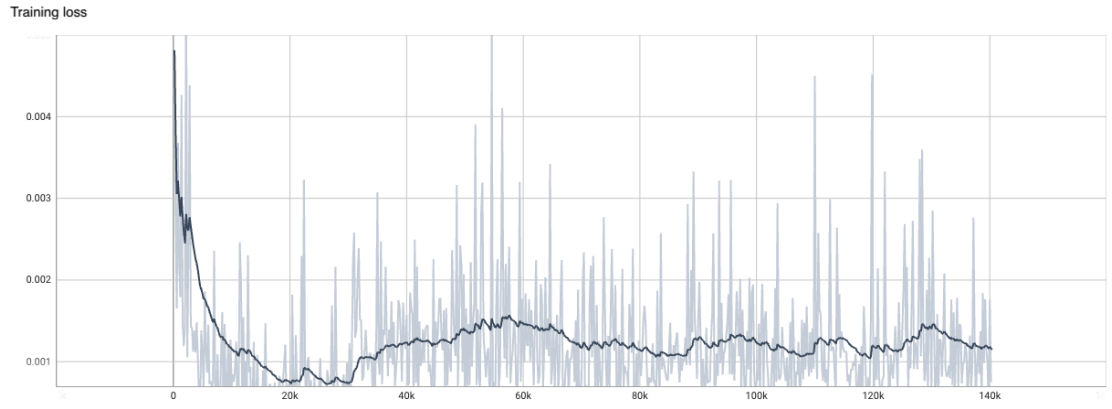


FIGURE 5.27: Learning Curve of 3ENeRF (Subject 2).

Eventually, the training progression of 3ENeRF is described in Figures 5.26 and 5.27. Our model exhibits a different trajectory compared to HyperNeRF and RigNeRF. Initially, the loss swiftly descends to a value beneath 0.001 within the first approximately 20,000 iterations. It then rebounds to values exceeding 0.001 after 30,000 iterations. This phenomenon can be traced back to our training strategy: the model was trained on the entire scene in the early iterations, which predominantly encompasses a static background. Given the inherent fixation of this background, it becomes relatively straightforward to learn, enabling the model to achieve a low loss when averaged across sampled points. Afterwards, the training swapped towards concentrating on the more complex dynamic facial region. As the deformation and appearance modules were not well-tuned for this facial region, there was an escalation in the loss. The loss then resumed gradually dropping and became stable at around 100,000 iterations, with additional training offering negligible enhancements. The oscillations observed in the learning curve of

our model suggest potential challenges in dealing with varying degrees of deformation across frames, possibly hinting at complexities arising from exaggerated deformations. Besides, its marginally greater final loss, in comparison to the other models, suggests a potential underfitting within our model. Nevertheless, an encouraging observation is the accelerated convergence rate exhibited by our model. Both static backgrounds and dynamic facial regions demonstrate rapid convergence. This efficient stabilization can be credited to our encoding methodology, specifically the system of looking up the nearest grids and updating their values. This localized approach, as opposed to overhauling the entire network, fosters stability and speeds up convergence. Based on these empirical observations, we set 100,000 iterations as the training duration for our model.

The duration of each iteration is another component that determines the training time of a model. Less time that an iteration takes denotes quicker computations for each batch of input, i.e. a batch of 3D points in this context. To illuminate these distinctions, we've logged the number of iterations processed per second for each method in table 5.6. This table also enumerates the total iterations each model requires for optimal performance and, by extension, the cumulative training time for each scene. From the table, HyperNeRF emerges as the swiftest, processing roughly two batches of input data every second. It is followed by our model, 3ENeRF, and RigNeRF, processing approximately 1.6 and 0.74 iterations per second respectively. Delving into the factors causing these variations, we discern that the computational demands of the head mesh look-up process when calculating the deformation of each point drive our model and RigNeRF slower than HyperNeRF. As for each sampled point, these models need to identify its nearest point on the head mesh as well as the distance to the nearest point to deduce the 3DMM deformation. Even with the integration of the KD-tree, the computation remains time-intensive. Each batch comprises thousands of points, where each point requires an $O(log_n)$ time complexity (with n being the total vertices in the mesh, i.e., 4069). This escalates the complexity for each iteration to $O(mlog_n)$ where m represents the number of sampled points. On the other hand, HyperNeRF eliminates the need for this look-up as it relies solely on MLPs for deformation computation. While it employs denser networks which take longer to process each input, its total per-iteration duration remains the shortest. When comparing RigNeRF and our model, both of which base their deformations on 3DMM, our model nearly doubles the processing speed per iteration. This is attributed to our incorporation of a multi-resolution hash encoding

combined with a lightweight MLP. This combination, being computationally lighter, facilitates swifter batch computations.

Models	Iterations ↓	Iter/sec ↑	Total(in hrs.) ↓
HyperNeRF	∼200k	∼2.08	∼26.7
RigNeRF	∼150k	∼0.74	∼56.3
3ENeRF(Ours)	∼100k	∼1.61	∼17.2

TABLE 5.6: Training and rendering efficiency of HyperNeRF, RigNeRF and 3ENeRF.

In brief summary, our model, 3ENeRF, emerges as the most time-efficient among all models because of its rapid convergence and relatively fast per-iteration duration. It requires roughly 17 hours of training for each scene, which is over three times faster than RigNeRF, its 3DMM-based counterpart, and 1.5 times quicker than HyperNeRF while allowing users to modify facial expressions and head poses explicitly. This efficiency underscores our model's potential for real-world applications.

Chapter 6

Conclusion

In this research, we introduced a method for rendering 3D human portraits within real-world scene contexts built upon the integration of 3DMM and NeRF. Our method not only facilitates the manipulation of scene viewpoints but also empowers the modifications of the facial expressions and head poses of human heads. Three major contributions resulting from our research are outlined below:

1. **Innovative Data Collection and Processing Method.** We have presented a data collection and processing method that effectively resolves the alignment between the 3DMM coordinate and the world coordinate. As far as we know, there is no work that discusses how to place 3DMM predicted head meshes into the coordinate aligned with the background (i.e. the world coordinate), nor are there any existing datasets published for head meshes in real-world scenarios with multi-view information. Furthermore, our data collection and processing method enables capturing subjects in diverse environments, such as homes, classrooms, or theoretically even outdoor spaces. Our data collection and processing approach, which includes two-stage filming, triangulation, transformation by determining the x-y scale factor and z translation factor between the DECA coordinate and the camera coordinate, and finally fine-tuning with gradient descent based on the 2D images from 3DMM, is pioneered. Our experiments validate the ability of our methodology to accurately position the human head mesh within the world coordinate, which ensures that the deformation process occurs in the same coordinate where points sampling and rendering take place.

2. **Fresh Model for Portrait Generation.** Building upon our collected data, we have proposed a model capable of generating novel portrait images of specific identities against arbitrary backgrounds while affording control over their head pose, expression, and 3D viewpoint. Through experimentation with different components and architectures, we develop an integrated model that effectively and efficiently learns the geometry, appearance, and dynamics of portraits. Drawing inspiration from existing work, we incorporate several additional methodologies such as scene normalization and contraction to enhance the adaptability of our model.

3. **Efficient Training with Multi-Resolution Hash Encoding Integration.** Initially, we trialled a fully MLP-based model but discovered that training such as a model will take three to four days, impeding it from deploying in real-world applications. In response, we integrate a multi-resolution hash encoding method introduced by Instant-NGP [11], which proves to significantly improve training efficiency while maintaining similar rendering outcomes. To the best of our knowledge, we are the first to incorporate the multi-resolution hash encoding into rendering human portraits with full control over expression, head pose, and viewpoint, while capturing background information from various viewpoints.

Overall, our research expands the capabilities of NeRF and facilitates the generation of controllable portrait images in diverse real-world scenarios, presenting a novel and comprehensive approach to portrait generation. Our experiments validate the ability of our methodology to accurately position the human head mesh within the world coordinate, which ensures that the deformation process occurs in the same coordinate where points sampling and rendering take place. Furthermore, the rendering results demonstrate that our model can render portraits that are versatile in terms of viewpoints and facial appearances. The training durations prove that our model is faster to be optimized compared to other examined models.

6.1 Limitations and Future work

While our research introduces a new data processing technique and a model with efficient training capabilities, there are certain limitations. One constraint observed is

the less-than-optimal rendering performance when compared to models like HyperNeRF and RigNeRF as shown in section 5.5. This limitation may stem from our choice of the backbone model for accelerated training: the multi-resolution hash encoding. Given that portrait rendering depends on a multitude of factors, it is plausible that a lightweight MLP may not fully capture all intricacies. To address this constraint, future work will involve extensive testing and experiments, especially concerning the integration of 3DMM with other efficient NeRF architectures. Models like DONeRF [46] and AdaNeRF [47] are of particular interest. These models optimize the training efficiency via decreasing the number of sampled points while maintaining a dense rendering network. This approach could reduce the computational burden of 3DMM deformation by calculating fewer points to the nearest mesh vertices and distances to those vertices, and it also ensures that the rendering network is robust enough to handle the comprehensive information of view direction, current facial expression, pose and so on.

In addition, as elucidated in section 5.5.3, when we try to reenact the facial actions of one person on the face of another, it is hard for us to observe a perfectly reproduced face on that person's face, especially for finer details such as the degree of mouth opening and features such as teeth. One contributing factor is the potential bias between the 3DMM-predicted meshes and reality, and another may be that we rely on meshes to infer deformation and on expression and pose parameters to infer appearance, which do not encapsulate fine details. In the upcoming research, we will further explore and exploit the 3DMM. The currently applied model, DECA, contains a "detail model" that enables DECA to capture pores, moles, expression-related wrinkles and other facial details by generating a UV displacement map. In the future, we will integrate this UV displacement map into our model. For example, for each sampled 3D point, we can first determine which vertex in the 3D mesh it is most proximate to and subsequently locate the corresponding displacement value for that mesh vertex by converting the UV map to 3D space. The UV displacement value can then be concatenated to the input of the networks to give them a semantic description of what the details should look like for a particular expression. For modelling accuracy, other 3DMMs like MICA [25] outperform DECA in the 3DMM public benchmark, suggesting they could yield more accurate representations of human heads. Similar to DECA, MICA employs FLAME as the underlying geometric model. Since MICA fine-tuned the FLAME model using its 3D annotated dataset and a designed loss function, their facial shapes may be modelled

more accurately, suggesting that the 3DMM deformation computed with the meshes inferred by MICA may be more precise than the ones inferred by DECA.

Another limitation pertains to the whole-image rendering efficiency of our model. While our approach speeds up training, the time required to render an image is still on par with that of other models, often exceeding one minute. This bottleneck arises because rendering a complete image necessitates processing a significantly larger number of points compared to the training phase. Consequently, the time taken with the 3DMM deformation computation adds substantial overhead. The refined sampling strategy mentioned in the previous paragraph provides one way to improve, while another promising direction is to improve the strategy for locating the nearest points and calculating their distances. In our current model, we employ the method of constructing a KD tree of the head grid at each frame to facilitate an efficient nearest-neighbour search. Although this approach outperforms linear search techniques, it still falls behind other more optimized approaches, especially those deployable on GPUs. In future research, we aim to replace the KD-tree approach with techniques such as Boundary Volume Hierarchy (BVH) [50] which enables parallel nearest-triangle searches on GPUs, thus further shortening rendering durations.

In future work, we intend to introduce additional loss terms to enhance the training process and avoid local optima. In our current approach, head meshes are only employed for deformation calculations. Given that the 3DMM meshes exist in the world coordinate, they can act as a valuable priori to guide the learning of the point densities in the scene. If a sampled point coincides with a mesh vertex, its density should be trained to 1. Conversely, if the point is located around the mesh and does not intersect with the background (we can use COLMAP to obtain the coarse underlying geometry of the background), the density should be trained to 0. Introducing such a loss term promotes more accurate learning of the true geometry, mitigating the problems associated with local suboptimization or overfitting of the wrong geometry.

Bibliography

[1] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, page 187–194, USA, 1999. ACM Press/Addison-Wesley Publishing Co. ISBN 0201485605. doi: 10.1145/311535.311556. URL https://doi.org/10.1145/311535.311556.

[2] Tianye Li, Timo Bolkart, Michael. J. Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4D scans. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6):194:1–194:17, 2017. URL https://doi.org/10.1145/3130800.3130813.

[3] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets, 2014. URL https://arxiv.org/abs/1411.1784.

[4] 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. In *ECCV*, 2020.

[5] 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.

[6] Sida Peng, Junting Dong, Qianqian Wang, Shangzhan Zhang, Qing Shuai, Xiaowei Zhou, and Hujun Bao. Animatable neural radiance fields for modeling dynamic human bodies. In *ICCV*, 2021.

[7] Wei Jiang, Kwang Moo Yi, Golnoosh Samei, Oncel Tuzel, and Anurag Ranjan. Neuman: Neural human radiance field from a single video, 2022. URL https://arxiv.org/abs/2203.12575.

[8] ShahRukh Athar, Zexiang Xu, Kalyan Sunkavalli, Eli Shechtman, and Zhixin Shu. Rignerf: Fully controllable neural 3d portraits. In *Computer Vision and Pattern Recognition (CVPR)*, 2022.

[9] Jingxiang Sun, Xuan Wang, Yong Zhang, Xiaoyu Li, Qi Zhang, Yebin Liu, and Jue Wang. Fenerf: Face editing in neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7672–7682, 2022.

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

[11] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, July 2022. doi: 10.1145/3528223.3530127. URL https://doi.org/10.1145/3528223.3530127.

[12] Xiang Guo, Guanying Chen, Yuchao Dai, Xiaoqing Ye, Jiadai Sun, Xiao Tan, and Errui Ding. Neural deformable voxel grid for fast optimization of dynamic view synthesis. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, 2022.

[13] Wojciech Zielonka, Timo Bolkart, and Justus Thies. Instant volumetric head avatars. *Conference on Computer Vision and Pattern Recognition*, 2023.

[14] Bernhard Egger, William A. P. Smith, Ayush Tewari, Stefanie Wuhrer, Michael Zollhoefer, Thabo Beeler, Florian Bernard, Timo Bolkart, Adam Kortylewski, Sami Romdhani, Christian Theobalt, Volker Blanz, and Thomas Vetter. 3d morphable face models—past, present, and future. *ACM Trans. Graph.*, 39(5), jun 2020. ISSN 0730-0301. doi: 10.1145/3395208. URL https://doi.org/10.1145/3395208.

[15] ShahRukh Athar, Zhixin Shu, and Dimitris Samaras. Self-supervised deformation modeling for facial expression editing. In *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)*, pages 294–301, 2020. doi: 10.1109/FG47880.2020.00115.

[16] A. Pumarola, A. Agudo, A.M. Martinez, A. Sanfeliu, and F. Moreno-Noguer. Ganimation: One-shot anatomically consistent facial animation. 2019.

[17] A. Tewari, J. Thies, B. Mildenhall, P. Srinivasan, E. Tretschk, W. Yifan, C. Lassner, V. Sitzmann, R. Martin-Brualla, S. Lombardi, T. Simon, C. Theobalt, M. Nießner, J. T. Barron, G. Wetzstein, M. Zollhöfer, and V. Golyanik. Advances in neural rendering. *Computer Graphics Forum*, 41(2):703–735, 2022. doi: https://doi.org/10.1111/cgf.14507. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.14507.

[18] Guy Gafni, Justus Thies, Michael Zollhöfer, and Matthias Nießner. Dynamic neural radiance fields for monocular 4d facial avatar reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8649–8658, June 2021.

[19] Kacper Kania, Kwang Moo Yi, Marek Kowalski, Tomasz Trzciński, and Andrea Tagliasacchi. Conerf: Controllable neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18623–18632, June 2022.

[20] Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Niessner. Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[21] Yu Deng, Jiaolong Yang, Dong Chen, Fang Wen, and Xin Tong. Disentangled and controllable face image generation via 3d imitative-contrastive learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[22] ShahRukh Athar, Zhixin Shu, and Dimitris Samaras. Flame-in-nerf: Neural control of radiance fields for free view face animation, 2021. URL https://arxiv.org/abs/2108.04913.

[23] Yufeng Zheng, Victoria Fernández Abrevaya, Marcel C. Bühler, Xu Chen, Michael J. Black, and Otmar Hilliges. I M Avatar: Implicit morphable head avatars from videos. In *Computer Vision and Pattern Recognition (CVPR)*, 2022.

[24] Yao Feng, Haiwen Feng, Michael J. Black, and Timo Bolkart. Learning an animatable detailed 3D face model from in-the-wild images. *ACM Transactions on Graphics, (Proc. SIGGRAPH)*, 40(8), 2021. URL https://doi.org/10.1145/3450626.3459936.

[25] *Towards Metrical Reconstruction of Human Faces*, 2022.

[26] Anurag Ranjan, Timo Bolkart, Soubhik Sanyal, and Michael J. Black. Generating 3d faces using convolutional mesh autoencoders. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[27] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Commun. ACM*, 63(11):139–144, oct 2020. ISSN 0001-0782. doi: 10.1145/3422622. URL https://doi.org/10.1145/3422622.

[28] Marek Kowalski, Stephan J. Garbin, Virginia Estellers, Tadas Baltrušaitis, Matthew Johnson, and Jamie Shotton. Config: Controllable neural face image generation. In *European Conference on Computer Vision (ECCV)*, 2020.

[29] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[30] Peter Hedman, Pratul P. Srinivasan, Ben Mildenhall, Jonathan T. Barron, and Paul Debevec. Baking neural radiance fields for real-time view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5875–5884, October 2021.

[31] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10318–10327, June 2021.

[32] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12959–12970, October 2021.

[33] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M. Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *ACM Trans. Graph.*, 40(6), dec 2021.

[34] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6498–6508, June 2021.

[35] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. Dynamic view synthesis from dynamic monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5712–5721, October 2021.

[36] Yilun Du, Yinan Zhang, Hong-Xing Yu, Joshua B. Tenenbaum, and Jiajun Wu. Neural radiance flow for 4d view synthesis and video processing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[37] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Smpl: A skinned multi-person linear model. *ACM Trans. Graph.*, 34(6), nov 2015. ISSN 0730-0301. doi: 10.1145/2816795.2818013. URL https://doi.org/10.1145/2816795.2818013.

[38] Lingjie Liu, Marc Habermann, Viktor Rudnev, Kripasindhu Sarkar, Jiatao Gu, and Christian Theobalt. Neural actor: Neural free-view synthesis of human actors with pose control. *ACM Trans. Graph.(ACM SIGGRAPH Asia)*, 2021.

[39] Sida Peng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. In *CVPR*, 2021.

[40] Junshu Tang, Bo Zhang, Binxin Yang, Ting Zhang, Dong Chen, Lizhuang Ma, and Fang Wen. Explicitly controllable 3d-aware portrait generation. *arXiv preprint arXiv:2209.05434*, 2022.

[41] Heng Yu, Koichiro Niinuma, and Laszlo A Jeni. Confies: Controllable neural face avatars. *arXiv preprint arXiv:2211.08610*, 2022.

[42] Tadas Baltrušaitis, Peter Robinson, and Louis-Philippe Morency. Openface: An open source facial behavior analysis toolkit. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–10, 2016. doi: 10.1109/WACV.2016.7477553.

[43] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5459–5469, June 2022.

[44] Liwen Wu, Jae Yong Lee, Anand Bhattad, Yuxiong Wang, and David Forsyth. Diver: Real-time and accurate neural radiance fields with deterministic integration for volume rendering, 2021.

[45] Tao Hu, Shu Liu, Yilun Chen, Tiancheng Shen, and Jiaya Jia. Efficientnerf efficient neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12902–12911, June 2022.

[46] Thomas Neff, Pascal Stadlbauer, Mathias Parger, Andreas Kurz, Joerg H. Mueller, Chakravarty R. Alla Chaitanya, Anton S. Kaplanyan, and Markus Steinberger. DONeRF: Towards Real-Time Rendering of Compact Neural Radiance Fields using Depth Oracle Networks. *Computer Graphics Forum*, 40(4), 2021. ISSN 1467-8659. doi: 10.1111/cgf.14340. URL https://doi.org/10.1111/cgf.14340.

[47] Andreas Kurz, Thomas Neff, Zhaoyang Lv, Michael Zollhöfer, and Markus Steinberger. Adanerf: Adaptive sampling for real-time rendering of neural radiance fields. 2022.

[48] Jia-Wei Liu, Yan-Pei Cao, Weijia Mao, Wenqiao Zhang, David Junhao Zhang, Jussi Keppo, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Devrf: Fast deformable voxel radiance fields for dynamic scenes. *arXiv preprint arXiv:2205.15723*, 2022.

[49] Tianjian Jiang, Xu Chen, Jie Song, and Otmar Hilliges. Instantavatar: Learning avatars from monocular video in 60 seconds. *arXiv*, 2022.

[50] James H. Clark. Hierarchical geometric models for visible surface algorithms. *Commun. ACM*, 19(10):547–554, oct 1976. ISSN 0001-0782. doi: 10.1145/360349.360354. URL https://doi.org/10.1145/360349.360354.

[51] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18(9):509–517, sep 1975. ISSN 0001-0782. doi: 10.1145/361002.361007. URL https://doi.org/10.1145/361002.361007.

[52] Thomas Müller. tiny-cuda-nn, 4 2021. URL https://github.com/NVlabs/tiny-cuda-nn.

[53] J.L. Pech-Pacheco, G. Cristobal, J. Chamorro-Martinez, and J. Fernandez-Valdivia. Diatom autofocusing in brightfield microscopy: a comparative study. In *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, volume 3, pages 314–317 vol.3, 2000. doi: 10.1109/ICPR.2000.903548.

[54] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. 24(6), 1981. ISSN 0001-0782. doi: 10.1145/358669.358692.

[55] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. doi: 10.1109/TIP.2003.819861.

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