Neural Radiance Field Driven Efficient and Controllable 3D portrait generation

Kuoyuan Li (1072843)

Supervisors:
Dr Mingming Gong
Dr Kris Ehinger
Dr Qiuhong Ke

Research Proposal

September 2022

Master of Computer Science
School of Computing and Information Systems
The University of Melbourne

Contents

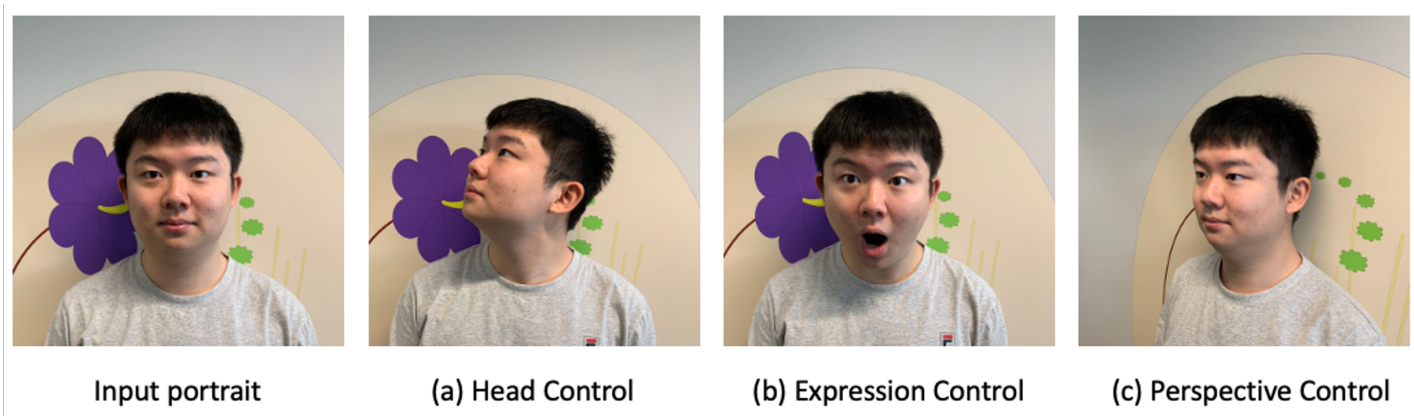

Figure 1: Given a human portrait, we want to synthesis new images with fully control over the (a) head pose, (b) expression and (c) perspective, while maintaining realistic.

1 Introduction

Faithful construction and manipulation of human portrait is a continuous research problem in the computer graphics and vision society. Given a human portrait, it is desirable to enable fully modification of its certain properties, such as 3D viewpoint, facial expression and head pose, while maintaining photo-realistic and high fidelity (Figure 1). Such functionalities present significance to the fields of augmented reality (AR), virtual reality (VR), and metaverse where human portraits are expected to be generated by a computer and can be controlled by users as desired. In these applications, users may anticipate their facial expressions reacting to the scenarios accordingly, or wish to reenact the behaviour of others. The generated portraits need to maintain reality to provide a 3D immersive experience. It is laborious and impractical to edit each individual manually whenever expected. However, automatically generate novel human portrait is challenging and remains unsolved despite the long-term attentiveness and recently growing research interest: the appearance of a portrait is not only impacted by the underlying facial geometry, but also other factors such as light conditions and skin reflectance under varying view points. Besides, portraits usually involve elements other than the face, such as accessories, hair, upper torso, and background.

In computer vision, the information required to render a portrait, such as its geometry, texture, and appearance, is commonly encapsulated in a model. Users could query and manipulate the model to render customized portraits. Over the past decades, different classes of models have been introduced. The 3D Morphable Face Model (3DMM) is an early approach to building explicit portrait models. It parameterises the human head and enables free manipulation of face shape, facial expressions and appearance [1]. Generative Adversarial Networks (GANs) have also been adapted for controllable generation of novel human portraits [2, 3]. However, when users attempt to control the viewpoints, their results have artefacts and are far from optimal, inspiring further approaches. Recent work focuses on rendering portraits with neural rendering models. Neural rendering combines insights from physics-based computer graphics and advances in deep learning. They leverage neural networks to represent complex scenes and enable production of photo-realistic images in a controllable manner [4]. Moreover, the freshly proposed Neural Radiance Field (NeRF) [5] demonstrates its surprising achievability

in learning 3D representations of complex scenes and objects. NeRF utilizes a neural network to derive volumetric representations of scenes without explicitly modelling 3D scenes with voxels, point clouds or meshes. It can synthesize photo-realistic images and videos under unseen camera positions and views. In the last two years, multiple models extend the original NeRF to render human portraits and empower users to control certain properties of them [6, 7, 8, 9]. These models are named as Editable NeRFs. The existing editable NeRFs are still far from perfect. Current models only generate portraits with low resolution (e.g. 256×256) [6] and require long training and inference times, which makes them burdensome to deploy on real world applications.

In this research, we aim to propose a new Editable NeRF model that can render portrait images in real-time with fine details. The input to the proposed model should be a monocular video that records the portrait of a specific person from different perspectives. After the model is trained, users could freely control certain attributes of the portrait such as expression, pose and viewpoints. The model can generate high-resolution images with required variations in real time.

2 Related Work

Our research aims to implement a model that enables fully control over head pose, expression and viewpoint while synthesising a human portrait, as well as maintaining fine-grained details and with high efficiency. This study is highly related to preceding studies on neural scene representations, 3D face modelling and novel view synthesis via Neural Radiance Field (NeRF).

2.1 3D Morphable Face Models

A collection of work exploits the parameters derived from 3D Morphable Face Models (3DMMs) to guide the novel human portraits generation [10, 11, 12, 13, 14, 15]. 3DMMs were firstly introduced by Vetter and Blanz [16] to generalize face representation by encoding the shape and texture of each face 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 shapes or textures from a massive number of 3D face scans. 3DMMs allow us to generate 3D face models with unseen shape and texture via manipulating these parameters (Figure 2). In the past several years, with the rise of neural networks, 3DMMs are studied in the context of deep learning [1, 17, 18, 19, 20]. These models are trained with larger datasets and provide multiple extended features such as modelling key articulations, eyeballs rotation [17], wrinkles [18], metrical shape [19] or extreme expressions [20]. 3DMMs could accurately parameterize human faces into low dimensions and enable free manipulations of certain properties like expression, texture, and pose. However, they only coarsely model faces without detailed texture information. They fail to incorporate other elements such as hair, teeth, and background as well. Multiple neural rendering models [10, 11, 14, 15] extend the concepts of 3DMMs and attach additional modules to enable the input portrait to reenact the actions or speeches of other individuals while maintaining the background and other details. These models could achieve controllable portrait generation in terms of expressions, poses, mouth shapes and appearance with

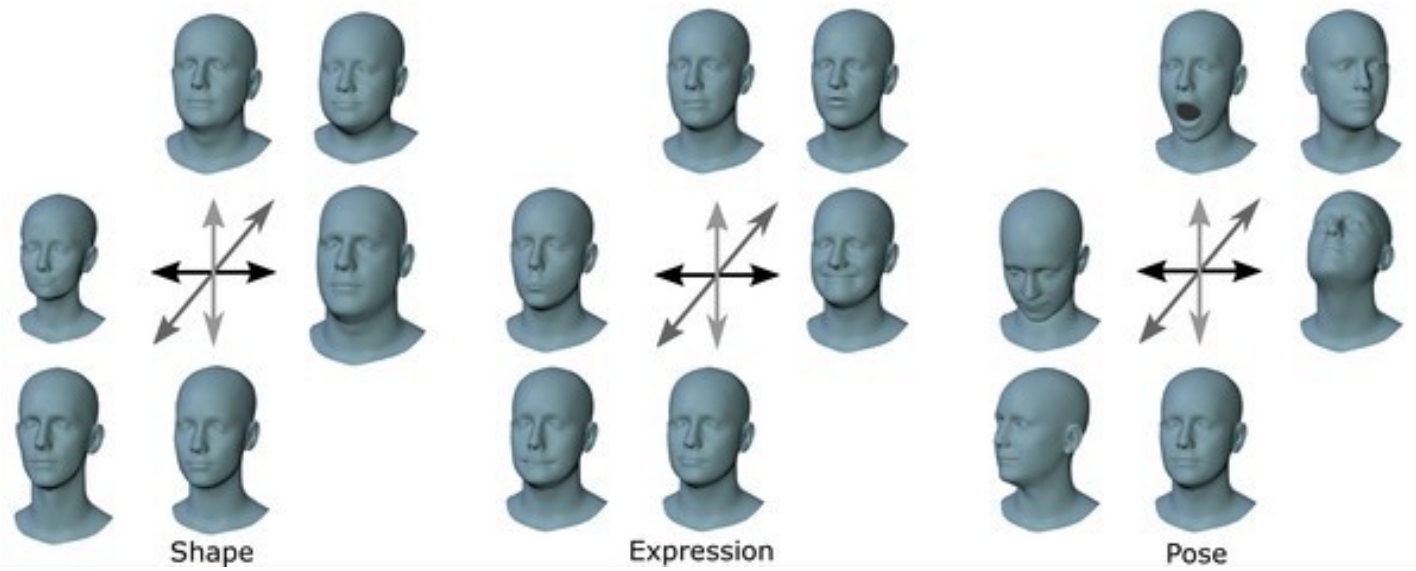

Figure 2: 3DMM demo. 3DMM decomposes human head into different properties. Varying the parameters for each property could result in different portraits.
Figure source: [17]

satisfying quality, but failed to generate portraits under novel camera perspectives. Furthermore, their performance relies on the underlying 3DMMs which usually ignore some properties of human portraits such as hair and mouth interiors, influencing the quality of eventual outcomes.

2.2 Generative Adversarial Networks

Deep generative models have been vastly adapted to conduct human portraits synthesis. Numerous models are built upon Generative Adversarial Networks (GANs) [21, 22] to generate human face images based on specified properties like muscle movements [2, 3], hairstyles [23], face shape, expression, and appearance [13, 24, 12]. These models typically decompose a portrait into different attributes (e.g. expression, hair, eye and mouth) and synthesis new images conditioned on these values. Some models delegate the decomposition to a 3DMM model [12, 13] to acquire low-dimensional feature representations while others are trained from scratch with large human face datasets. The generated images are justified by a discriminator to obtain an adversarial loss, which guides the generator to create more natural images. However, these models are inherently image-based and lack explicit 3D representation, making it challenging to control geometry properties and alter viewpoints freely [6].

2.3 Neural Radiance Fields

Neural radiance fields (NeRF) has achieved astonishing results on novel view generation for static scenes [5]. Given a short video of a single scene with known camera poses, NeRF optimizes a multilayer perceptron (MLP) to learn the volumetric representation of the scene. The MLP aims to accurately predict the volume density (σ) and color (RGB) of each 3D point (x, y, z) in the scene based on its location and view direction

(denoted by θ and ϕ). The density of each point σ determines how much radiance is accumulated passing through the point. For a ray with direction $\mathbf{d}$ and origin $\mathbf{x}$ (which can be computed from camera poses), NeRF firstly samples points in the ray, then feed these 3D points locations along with their view directions into the MLP to obtain their predicted colours and densities. NeRF 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. The fine network is used in the final rendering as it learns from more informative points. Such a process is named the hierarchical sampling procedure [5]. Each ray corresponds to one pixel in the rendered image. The eventual colour of the pixel is computed via integrating all sampled points along the ray with the classical volume rendering formula as shown in equation 2. In addition, NeRF introduces positional encoding (γ) which maps input 5D vector (x, y, z, θ, ϕ) into higher dimensions to further improve the fidelity of images with high-frequency variations. The coarse and fine MLPs are trained simultaneously to minimize the MSE loss between the ground truth image and the predicted image. Since the classical volume rendering function is intrinsically differential, the parameters in the MLPs can be optimized with gradient descent algorithms. In summary, the NeRF can be represented as:

$$c_i(\mathbf{x_i}, \mathbf{d_i}), \sigma_i(\mathbf{x_i}) = F(\gamma(\mathbf{x_i}), \gamma(\mathbf{d_i})) \quad and \tag{1}$$

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i(1 - exp(-\sigma_i\delta_i))\mathbf{c_i}, \text{ where } T_i = exp(-\sum_{j=1}^{i-1}\sigma_i\delta_i), \tag{2}$$

where F is the MLP to interfere the color c_i and the density σ_i given a particular 3D location $\mathbf{x_i}$ and the 2D view direction $\mathbf{d_i}$. In the original NeRF, only colour depends on the view direction since the density is a physical characteristic and remains constant under different view directions. δ_i is the distance between adjacent samples. N is the number of sampled points along a ray. $\hat{C}$ is the final colour of the ray which corresponds to one pixel.

The MLPs in NeRF implicitly store the 3D geometry and appearance of the scene which is relatively small compared with explicit modelling methods like dense voxel grid [25]. Despite realistic novel view generation, the vanilla NeRF suffers from the following issues: 1) It is dedicated to static scenes and cannot be directly applied on dynamic scenes. 2) The MLP only captures the information for one particular scene and requires a long training time. While rendering a new image, high volume of 3D points need to be sampled and the MLP is enquired tremendous times, implying difficulties on deploying to real-world applications where time is critical.

2.4 Dynamic Neural Radiance Fields

A family of dynamic NeRFs were initiated to handle objects motions and variations in different time frames [26, 27, 28, 29, 30, 31, 32]. D-NeRF [26], nerfies [27], HyperNeRF [28] and NR-NeRF [29] appended a deformation field that warps the deformed points into a canonical space before feeding them into the NeRF MLP. The points in the canonical space remain constant through time, hence satisfying the assumption by NeRF MLP: intersecting rays from different views are identical through time. The

points are deformed via adding a displacement value $\Delta(\mathbf{x})$ [26] or transforming with a dense SE(3) field [27, 28, 29] predicted by the deformation fields, which is a separate MLP that takes the 3D location of a point and its time step embedding as input values. Each time step embedding dedicates to one image and captures the geometry and appearance of it. The time step embedding, deformation MLP and NeRF MLP are optimized jointly to learn how to deform points at each moment and the volumetric representation of points simultaneously. Furthermore, additional losses are designed to avoid exaggerated deforming and boost the training efficiency [27, 29].

Similarly, Neural Scene Flow Fields [30] and Dynamic View Synthesis [32] warp 3D points, but instead of to a canonical space, they predicts a 3D motion vector describing how points move between the current time step and the neighbouring time steps. They devise additional regularization losses to learn how 3D points vary through the time and account for potential occlusions. Alternatively, Space-time Neural Irradiance Fields [31] claims that dynamics between different frames may be caused by appearance variation or 3D geometry variation, which cannot be distinguished in the vanilla NeRF and lead to artefacts when rendering objects motions. It utilizes a depth reconstruction loss to encourage the model to learn an accurate depth of each ray per time step, hence constraining the scene geometry at any time. Rendered images from unseen views could preserve reality as their geometry variations are learnt.

These dynamic NeRFs do not allow control over other properties in human portraits except time-dependent viewpoints. Nevertheless, they provided a foundation to handle motions in the scene and their architectures are widely adapted in editable NeRFs [33, 6, 7].

2.5 Efficient Neural Radiance Fields

The vanilla NeRF requires heavy computing power to train and render. It takes approximately two days to train a NeRF model with a high-performance GPU [5]. Moreover, rendering each frame requires querying the MLP hundreds of times and takes around half a minute. Rendering a high-resolution video with commercial hardware may take hours. Such issues prohibit NeRF from deploying on real-world applications where time is critical. To address them, variants with high efficiency and fast rendering are introduced. Many models sacrifice storage space to obtain faster rendering during inference. Models generally adapt voxel fields [34, 25, 35, 36, 37] or neural point clouds [38] to store the information of a 3D scene where each voxel or point is trained to encode a feature vector capturing the region-specific information like opacity, appearance and geometry of a point in the 3D world. These feature vectors are stored in optimized data structures such as trees [34, 37] or dense arrays [25] to save storage space and boost retrieval efficiency. The feature vector can be considered as an intermediate value that will be fed to a lightweight and well-trained MLP to acquire the view-dependent information [25, 35] or colour and opacity values [34, 36]. In addition, only voxels that correspond to non-empty parts of a scene are stored, i.e. where the density values are larger than zero. Points with zero density do not contribute to the final rendering and skipping such points could speed up the rendering during generation of new views.

Some work reduces both training and inference time by adjusting the hierarchical sampling procedure in the vanilla NeRF [37, 35, 39, 40]. E-NeRF [37] and DVGO [35] leverage voxel grids to record the density of each point during the training process.

The densities in the voxel grid are continuously updated until optimal during training. During the coarse sampling phase, instead of sampling points along the ray uniformly, non-contributing points (i.e. points with opacity close to or at zero) are omitted [37] or points are weighted sampled according to their densities [35]. During the fine sampling phase, only pivotal points, which are defined as points with densities greater than a preset threshold, are sampled to encourage the fine MLP focus on significant points. On the other hand, AdaNeRF [40] and DoNeRF [39] abandon the hierarchical sampling procedure and instead train dual networks, where a sampling network learns how to efficiently sample sparse and meaningful points along the ray and a shading network learns how to render based on the sparsely sampled points. The sampling and shading networks are jointly optimized by rendering losses and additional losses to encourage sparsity in sampling [40]. These efficient NeRFs are established based on the vanilla NeRF, and thus 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.

HeadNeRF [41] improves the training efficiency for human portraits via pre-training the MLP with human head datasets. Similar to 3DMM, HeadNeRF disentangles the human head into four factors: identity, expression, albedo and illustration. While rendering novel views of an unseen portrait, it does not need to be retrained, but only requires the extracted factors of this portrait and a particular perspective. However, HeadNeRF does not speed up the rendering as it still involves tremendous queries to the MLP. In addition, it only considers the human head and ignores other elements such as backgrounds.

2.6 Editable Neural Radiance Fields for Human Portraits

Although some of the above dynamic NeRFs [27, 28, 31] demonstrate their applicability in generating dynamic portraits, they do not support free control over head pose, facial expression and other details. To meet the requirement of full control, a family of editable NeRFs have been proposed [6, 33, 8, 9, 42, 7]. Most of the existing models are based on 3DMMs to achieve free manipulation of portraits [6, 33, 7], excluding GAN-based model [9] or models trained from scratch [8, 42].

FLAME-in-NeRF [33] achieves free manipulation of face expressions via altering expression parameters extracted with FLAME, which is a sub-model in the 3DMM DECA [18]. Similar to nerfies [27], it employs a deformation fields to warp points into a canonical space. For rays within the face, the expression parameters are appended during training and prediction. The silhouettes provided by FLAME are used to determine which rays are inside the face. The deformation MLP attempts to learn how to deform points to the canonical space based on known expression parameters. Hence when generating a new portrait, expressions can be engineered by simply modifying the expression parameters. In addition, FLAME-in-NeRF includes a learnable deformation and appearance encoding to provide additional information. Similarly, NerFACE [7] adopts 3DMM from the face2face model [11] to extract expression and pose parameters. NerFACE does not apply a deformation MLP, but uses the pose parameters (rotation and translation) to warp points. The NeRF MLP in NerFACE conditions on the expression parameters to learn correct renderings under different expressions simi-

lar to FLAME-in-NeRF. However, NerFACE assumes a static camera and background, thus forbidding its from free view generation.

One of the issues with the above methods is that the rendered images have unnatural traces due to their overly simple models. In addition, they only allow control of expressions [33] or do not support new view generation [7]. The researchers further explored 3DMM and proposed RigNeRF [6]. RigNeRF not only allows controlling the expressions and views of the generated portraits, but also enables head pose control by adding head pose parameters extracted from the DECA's FLAME model. As with the models mentioned above, RigNeRF also deforms the rays into a canonical space before feeding into the NeRF MLP. The canonical space is defined by the frontal head pose and neutral expression of the FLAME model. In RigNeRF, the deformation is no longer predicted by a single MLP, nor by the pose parameter, but the sum of the 3DMM deformation field and a residual deformation estimated by the deformation MLP. RigNeRF further exploits DECA to enable free generation on new views, head poses, and expressions. One problem with the RigNeRF is its relatively low resolution (256×256), thus cannot capture details of human faces such as wrinkles and moles.

Unlike 3DMM-based editable NeRFs, FENeRF [9] adopts a GAN-like structure, which consists of a generator to predict the color and density of each 3D point, and a discriminator to distinguish whether the rendered portrait is realistic or not. To permit free editing of the faces, FENeRF introduces two latent codes to operate on shapes and textures. The shape code controls the geometry of portraits, while the texture code governs the appearance. The generator takes the latent code, point position and view direction as input to yield color and density. FENeRF utilizes a CNN-based discriminator to discern the fidelity of the rendered portrait and trains the generator to obfuscate the discriminator. However, FENeRF assumes a consistent view and is unable to generate portraits from other viewpoints.

AD-NeRF [42] proposes a model for synthesizing high-fidelity audio-driven portrait videos. AD-NeRF does not rely on any intermediate representation like 3DMM but applies the audio features extracted with deep-speech [43]. The audio features are attached to the input vector of the NeRF MLP. AD-NeRF can achieve high fidelity not only for the head but also for the upper body. AD-NeRF states that head motion does not always align with the torso. Thus it builds two separate MLPs to learn the volumetric representations of the head and torso separately. To handle the deformed head pose, AD-NeRF conducts a similar approach to NerFACE [7] , i.e., transforming the head into a canonical space based on the pose transformation matrix. New portrait videos can be generated by inputting new audio features and different head pose matrices. AD-NeRF still does not have full control over the expression except for the mouth shape based on the input audio. CoNeRF [8] further increases the freedom of controlling expressions by abandoning the underlying 3DMMs and training from scratch. It subdivides the expression control to individual attributes, e.g. an attribute can control whether the mouth is open or closed. Users can define a set of attributes as they wish and sparsely annotate the training data to indicate the regions affected by each attribute. After a short training period, the attribute extraction module in the model can automatically extract the attribute values from the unannotated images. These attribute values and latent codes will be processed by several other modules to eventually produce colour and density. While synthesis, we can manipulate the values of each attribute and feed them into the model to generate new expressions. CoNeRF introduces more modules,

which means more complex networks, longer training time and inference time. It requires several annotated images to train the attributes extractors at the beginning. These factors increase the difficulty of its application in practice. In addition, CoNeRF focuses on the expressions and does not generate new head poses.

2.7 Review Summary

Based on our review, NeRF can render high-quality images relying solely on a simple MLP. Many subsequent works have improved the performance of NeRF and its usage scenarios: numerous efficient NeRFs can be trained rapidly and render scenes in real-time; multiple editable NeRFs can control the generated portraits. Despite these extensions, to the best of our knowledge, existing editable NeRFs can only render low-resolution images without fine details due to long inference times. Our research will focus on addressing the research gap between efficient NeRFs and editable NeRFs, targeting to introduce editable NeRF models capable of rendering portraits in real-time.

3 Research Question

Question 1: How to extend the editable NeRF to capture the details of human portraits?

None of the existing Editable NeRFs can generate high-resolution portraits with fine detail while permitting entirely free manipulation. Some models can control face pose, expression and new viewpoints, but they cannot represent variations in the background under different viewpoints, thus generating portraits with a blank or consistent background [7, 9, 42]. RigNeRF [6] and CoNeRF [8] can capture changes in the background and other non-head parts from different viewpoints, and can also modify pose, expression [6] or predefined facial details [8] as illustrated above. However, their relatively low resolution makes them fail to catch details in and around faces, such as wrinkles, moles and textures on the surface of accessories. In addition, CoNeRF expects additional annotations to supervise the learning, making it more cumbersome to apply in real life. Therefore, we will improve the resolution and detail attaining capability of RigNeRF. RigNeRF only incorporates the FLAME model in DECA without further exploitation, even at its core: the ability to model wrinkles. With the development of 3DMM, more accurate models are introduced such as MICA [19]. These extraordinary face models are not exhaustively being studies by existing editable NeRFs. Our research aims to resolve this knowledge gap.

Question 2: How to improve the training and inference speed?

One of the main factors preventing precedent models from generating high-resolution images is the long rendering time. In NeRF, each pixel is rendered independently. The number of queries for MLP increases exponentially as the resolution increases. The discussed efficient NeRFs in section 2.5 seem to offer potential solutions: the trained feature vectors can be stored in a voxel grid or point cloud, which are available at rendering time to boost the efficiency. In addition, 3D representations of face models

have been extensively studied [17, 18, 44]. These models prove 3D representations like voxel field and mesh grid can be applied on portraits. However, these models are usually very large and does not capture details. We want to retain the core advantages of NeRF models which is relatively small in size and has good rendering quality at details, hence no explicit 3D models are stored, but just store have some additional voxel grids to improve the efficiency as in previous work.

4 Research Plan

4.1 Data collection and processing

NeRF learns a single scene directly and no large dataset is needed. The expected data to train our model is a portrait video of a particular person with different head poses and expressions, taken from different viewpoints. Early studies prove videos captured with mobile phones are sufficient to train NeRF models [5, 27, 6, 8, 28], therefore our videos will be recorded in HD mode with the iPhone XR, yielding a 1080×1920 video. Following the procedure in RigNeRF [6], in the first part of the filming, the subject is asked to make different expressions while keeping their head still. The camera will move around them to capture different viewpoints. In the second part, the camera is fixed in front of the head and the subject is asked to rotate their head and continue making expressions. In addition, the background should be static and contain rich textures to ensure that the camera pose can be estimated accurately. The length of the video is expected to be around one minute to contain adequate information. The video will be further processed into frames and blurred frames will be filtered out based on their Laplacian [45] variances, resulting in approximately 2,000 HD portrait images.

Similar as prior NeRFs [5, 26, 28, 6], we will use COLMAP [46] to identify the camera pose in each frame. COLMAP is a structure-from-motion (SfM) and multi-view stereo (MVS) model. It could extract camera poses from a collection of images of the same scene. To prevent COLMAP from matching feature points on moving parts, which causes misalignment of features and impacts the accuracy of estimated poses, we will follow nerfies [27] to introduce a face segmentation network that masks out the dynamic head part in images to constrain the COLMAP learning from static background.

To examine the robustness of our model, we will collect portrait videos from 5-10 volunteers from different ethnicities and genders under different scenes (indoor or outdoor) and lighting conditions. Each collected dataset will be randomly splitted into training and evaluation datasets, where 70% is for training and the remaining 30% is for evaluation. The evaluation dataset is unseen during training and used to quantitatively justify our model's performance. Evaluation metrics will be discussed in section 5.

4.2 Model Implementation and Experiment

Our work will build on the existing RigNeRF model [6] with adjustments and enhancements. Unfortunately, the RigNeRF team has not released their source code. We will reproduce its results via modifying the source code of nerfies [27], which is open-source and share many similarities with RigNeRF in architecture: they both contain a learn-

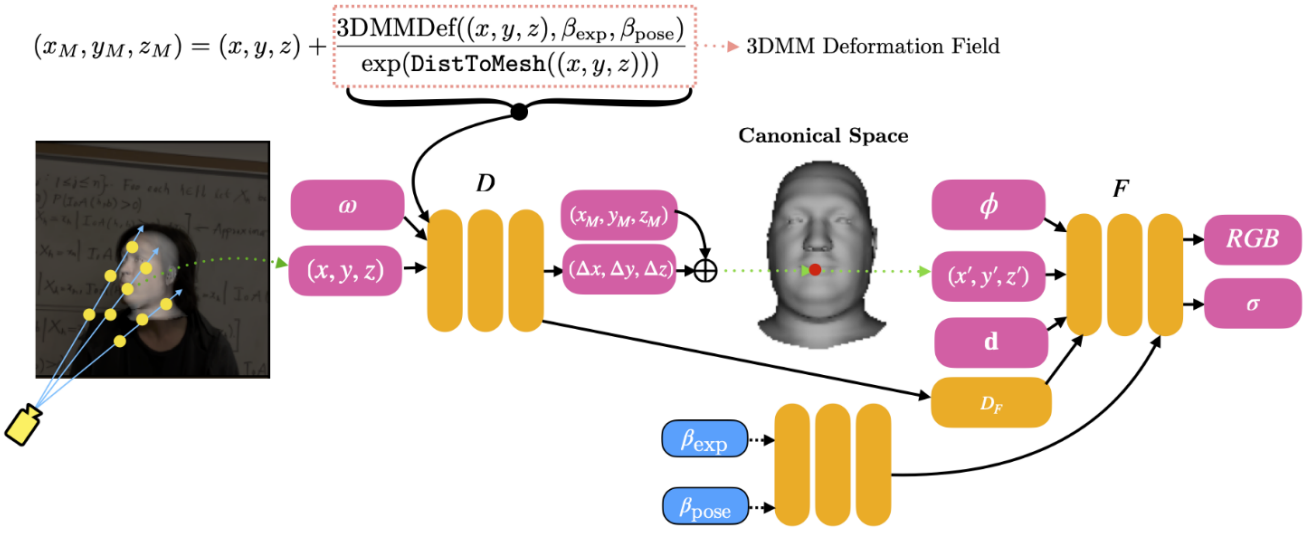

Figure 3: RigNeRF Architecture. Figure source: [6].

able deformation and appearance codes to account deformation and appearance variations over time. Nerfies is developed on JaxNeRF [47]. There are several additional components need to be appended on nerfies. As shown in figure 3, the deformation MLP in RigNeRF is conditioned on the 3DMM Deformation field in addition to the 3D position and deformation code. The 3DMM Deformation field can be computed directly from the DECA [18] model, which is open-source and off-the-shelf. For the NeRF MLP, the RigNeRF attaches the feature map from the penultimate layer of the deformation MLP and the expression and pose parameters derived from DECA to the input vector, encouraging the MLP to learn how the deformations should affect the appearance of the point.

Figure 4 shows one configuration of our model. We propose two changes to RigNeRF: extra 3DMM parameters for NeRF MLP and voxel field-oriented sampling and rendering. RigNeRF follows the hierarchical sampling procedure stated in the vanilla NeRF where many meaningless points are sampled and waste plenty of time. In our model, the sampling procedure will be guided by the voxel field where points with zero density are ignored. The deformation field is identical to RigNeRF which is the total of the FLAME-based 3DMM deformation field and a residual displacement deduced by the deformation MLP. It is unchanged because the deformation should theoretically be determined only by the head pose and expression variations, which can be adequately described by the FLAME model and a residual displacement from the MLP. The deformed 3D position is used to look up the corresponding stored values in the neural voxel field, which is fed into a lightweight MLP with auxiliary parameters to obtain the density and view-dependent color. Our experiments will examine the components in DECA besides from FLAME. Moreover, we will trial the State-of-the-Art 3DMM MICA [19] which surpasses DECA and leads the 3DMM benchmark. Different configurations of neural voxel fields will be evaluated to determine the best one for dynamic portrait. In the following sections, we will discuss how to experiment different 3DMMs and how to introduce voxel fields into our model in details.

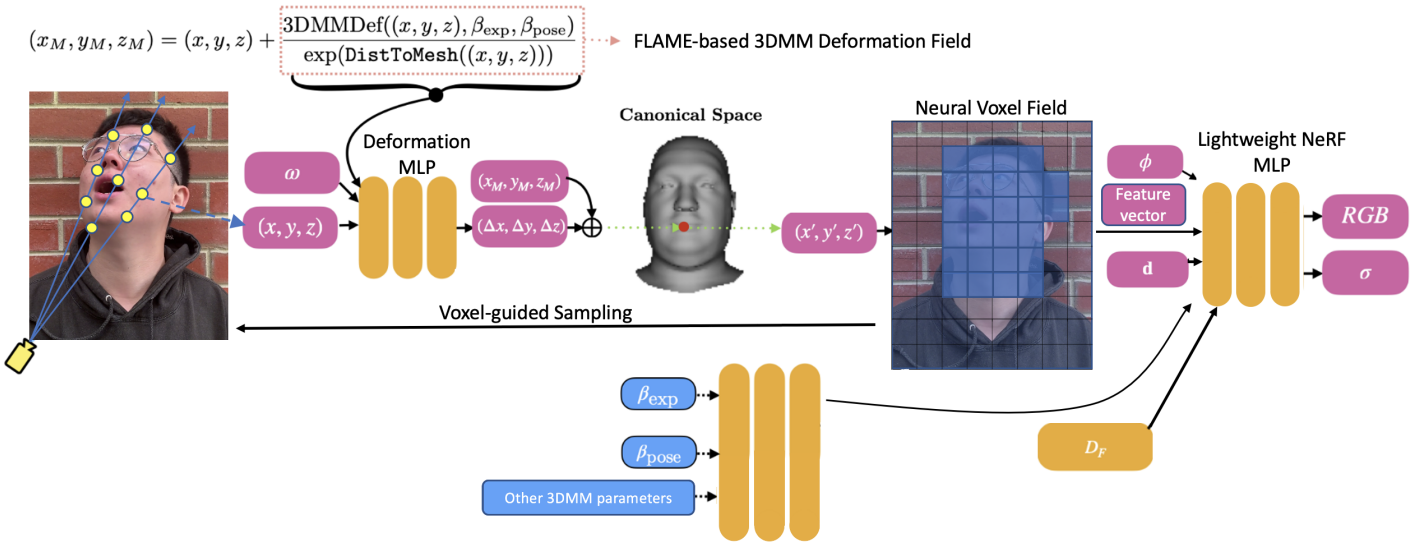

Figure 4: Proposed model Architecture. Based on a figure in [6].

4.2.1 3DMM Experiments

In addition to the geometric model FLAME, DECA contains an appearance model, a camera model and a detail model [18]. The appearance and camera models are devoted for images rendering. Since NeRF relies on a different rendering algorithm than DECA, trialing appearance and camera models are meaningless. Our focus will be on the detail model, which provides DECA with the capability to capture pores, moles, expression-related wrinkles and other facial details. The detail model generates a UV displacement map based on the expression and jaw pose parameters. In our model, for each sampled 3D point, we will firstly identify which point in the 3D mesh it corresponds to, then find the displacement value of that mesh point by converting the UV map into the 3D space. As DECA is trained with large datasets, this displacement value should semantically describe how the details below a particular expression should look like. Such information may be useful for depicting details in our high resolution images. The UV displacement values are shown as other 3DMM parameters in figure 4. The UV displacement map can be derived while running the entire DECA model for each frame. Therefore, no additional computational effort will be incurred compared with RigNeRF.

A new 3DMM model, MICA, surpasses DECA on the 3DMM public benchmark [19, 48], which implies its potential for more accurate computation of human expressions and head pose disparities. In contrast to DECA which is trained purely with 2D images, MICA is trained with 3D data and optimized to reconstruct the correct metrical shape of human faces. Similar to DECA, MICA use FLAME as the underlying geometric model. The 3DMM deformation field with MICA will be the same as it with DECA. Because MICA fine-tunes the FLAME model with their 3D annotated datasets and designed loss functions, it could more precisely model the facial shape, which indicates the 3DMM deformation field computed by MICA may be more precise than DECA in the facial region. We will experiment whether substitute DECA with MICA for the deformation field could give rise to superior results.

4.2.2 Voxel Fields Experiments

Majority of the reviewed efficient NeRFs store information under the notion of voxel fields, but their values in each voxel are distinct. We will experiment with different architectures. Since these efficient NeRFs are developed for static scenes, additional calibration is expected to accommodate our dynamic scenes. Voxels can purely store densities which are used to guide the sampling procedure [37, 35]. We will first investigate the effect of leveraging stored densities on our model in terms of speed and performance. A voxel field of dimension D×D×D is introduced where each voxel keeps track of the density of the point. Following E-NeRF [37], each point will be initialized with a small and identical value. During the training process, the density is updated based on the estimation of the NeRF MLP. When the densities stabilize after a few iterations, points with zero density are pruned to save storage space. During fine-tuning and inference, if sampled points do not exist in the voxel field, the density of this point is zero and it can be simply ignored.

Under such architecture, if the speed is boosted as in theory and the rendering quality is not inferior to the original setup, we will further explore the effect of storing additional values used to estimate the final colour of a ray at a faster rate, as in the previous work [25, 34, 36]. We will follow the architecture of SNeRG [25] since it demonstrates an outstanding rendering speed compared to other efficient NeRFs without losing much detail. The NeRF MLP will be trained to yield an additional 4D feature vector besides from the density and diffuse color of each point. These 3 values are stored in a voxel field and updated at each iteration. In addition, we will train a lightweight MLP to derive view-dependent appearance values. When rendering a ray with the optimized model, it does not feed the points into the NeRF MLP to get view-dependent colors, but accumulates diffuse colors and feature vectors from all points along the ray. The accumulated feature vector is sent to the trained lightweight MLP to obtain a view-dependent residual color and add to the accumulated diffuse color. In our model, the lightweight MLP, Nerf MLP and deformation MLP are trained simultaneously with the MSE loss. The NeRF MLP and lightweight MLP will be subject to extra 3DMM parameters as mentioned above. This architecture suggests that we only need to query the lightweight MLP once for each ray, rather than querying the large MLP multiple times during inference, which improves the rendering efficiency to a large extent.

One issue for efficient NeRFs (including E-NeRF and SNeRG) is that they assume a static scene. In dynamic scenes, the density of each point is inconsistent over time. For example, a moving head can cause some space to be occupied from vacant. The voxel field should store the density in a canonical space, which is defined as frontal head pose and neutral expression in our model. While we check the density of a point, we need to first deform it to the canonical space before looking up the voxel field to prevent missing meaningful points. We argue that the residual displacement predicted by the deformation MLP is very small. It is significant for detail rendering but may not be influential for density checking during sampling. Therefore, we will simplify the warping by using only the 3DMM deformation value, which is fast to compute and focuses on large variations. We will experiment with this setup to determine if our hypothesis is valid and if the rendering is not affected.

4.3 Training Details

The model will be trained on four NVIDIA P100 GPUs. We will adhere to the training setup of RigNeRF [6] initially and experimentally tune the hyperparameters. The number of sampled points per ray is initially 128, and is gradually decremented during the training process to accelerate the rate. For the position encoding, 10 frequencies are applied to encode the position and 4 frequencies are applied to encode the direction. The architectures of the deformation and NeRF MLPs are the same as RigNeRF, with deformation MLP being 8 layers of 128 channels each and NeRF MLP being 8 layers of 256 channels each. The lightweight MLP will follow the SNeRG [25] with 2 layer of 16 channels each. The learning rate will be initially 5e-4 and decay to 5e-5 at the end of training. The model will be trained for 150,000 epochs. These are initial settings and will be adjusted accordingly during our experiments.

5 Evaluation methods

5.1 Quantitative Metrics

The hold out frames from the captured videos will be used to quantitatively evaluate the performance of our models under different configurations and to compare against baseline models. Models will be measured in two aspects: accuracy and efficiency. For accuracy, four common metrics for NeRF models are Mean-Square Error (MSE), Peak Signal-to-Noise Ratio (PSNR), Structural Similarity (SSIM) [49] and Learned Perceptual Image Patch Similarity (LPIPS) [50]. RigNeRF additionally considers a FaceMSE, which is the MSE over the facial region only [6]. Since our focus is on portraits, FaceMSE is included as well. MSE gauges the average of the squared errors between the inferred image and the ground truth. A lower MSE indicates the rendered image is similar to the real image. Peak signal-to-noise ratio (PSNR) is calculated based on MSE and measures the quality of the reconstruction. A higher PSNR indicates a better reconstruction. Structural similarity index (SSIM) measures the similarity between two images, and higher values are preferred. Learning Perceptual Image Patch Similarity (LPIPS) determines the perceptual similarity between the inferred image and the actual image. A smaller LPIPS means the inferred image is more similar to the ground truth. HyperNeRF [28] claims that PSNR and MSE are sensitive to small offsets, while PSNR penalizes clear images over blurred images. SSIM may not detect obvious artifacts. Among them, LPIPS best describes perceptual quality. Therefore, we will emphasize on LPIPS when justifying models.

Because the running speed is hardware dependent, we will experiment with the same device for all models. We will track the training duration of models with the same number of epochs, measured in minutes where less minutes are more ideal. The rendering time will be gauged in frames per second (FPS), which describes how many images the model could render in one second. Higher FPS is more desirable. In addition, we will consider FPS per watt (FPS/W), which is widely adopted in the high-performance graphics community to measure the performance relative to power consumption [51]. A high FPS/W indicates the model can render images rapidly with low energy usage.

5.2 Baseline Models

We select RigNeRF [6], NerFace [7] and CoNeRF[8] as our baseline models because they allow users to freely edit certain properties of a particular portrait. Our primary focus is RigNeRF since our model builds on it. The experiment dataset will be the portraits we collected and the videos from CoNeRF. RigNeRF and NerFace have not publicly released their datasets. Since CoNeRF expects annotated images, we will manually label our frames as instructed in their documentation. We will compare our models with the baseline models using the above accuracy and efficiency metrics. In addition, to evaluate models' editing capabilities, since no ground truth images exist, we will qualitatively analyze the quality of rendered images under new views, specified expressions and poses. In particular, we will focus on details such as facial features, wrinkles and accessories texture, and observe whether obvious artefacts exist. Motivated by prior literature [6, 8, 7, 9, 5], we will demonstrate some instance frames derived from all models to assist the analysis and discussion.

6 Timeline

Please refer to the appendix for the landscape timeline.

Stage	Task	Number of weeks	Duration
Preparation	Data collection and processing	Done	Done
	Literature review and proposal writing	Done	Done
	Implementation of RigNeRF	3	12/9 - 9/10
Presentation	Oral Preseantation preparation	2	19/9 - 9/10
	Oral Preseantation	2	10/10 - 21/10
Experiment (Baseline)	Baseline models experiments on collected dataset	2	10/10 - 23/10
Experiments (3DMM)	DECA implementation and experiments	3	17/10 - 6/11
	MICA implementation and experiments	2	7/11 - 20/11
Experiments (Voxel Field)	Voxel field (Density) implementation and experiments	6	21/11 - 23/12
	Voxel field (Feature vector) implementation and experiments	8	2/1 - 26/2
Evaluation	Models evaluation and error analysis	3	27/2 - 19/3
	Models adjustment and finetune	6	27/2 - 9/4
Thesis	Thesis writing	10	19/3 - 4/6

Figure 5: Proposed Timeline.

7 Research implications and Social impacts

Our research aims to address the existing problems of editable NeRFs, concentrating on low resolution, blurred details and slow rendering speed. The improved quality and faster speed could further stimulate NeRF-based real-world applications for editing a specific portrait in domains like VR and AR. A potential negative social impact of our

model is that it can be misused by malicious individuals to produce deep fake images and videos. Existing discriminator networks can be used to detect synthetic images and avoid such problems.

References

[1] B. Egger, W. A. P. Smith, A. Tewari, S. Wuhrer, M. Zollhoefer, T. Beeler, F. Bernard, T. Bolkart, A. Kortylewski, S. Romdhani, C. Theobalt, V. Blanz, and T. Vetter, "3d morphable face models—past, present, and future," *ACM Trans. Graph.*, vol. 39, no. 5, jun 2020. [Online]. Available: https://doi.org/10.1145/3395208

[2] S. Athar, Z. Shu, and D. Samaras, "Self-supervised deformation modeling for facial expression editing," in *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)*, 2020, pp. 294–301.

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

[4] 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*, vol. 41, no. 2, pp. 703–735, 2022. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.14507

[5] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, pp. 405–421.

[6] S. Athar, Z. Xu, K. Sunkavalli, E. Shechtman, and Z. Shu, "Rignerf: Fully controllable neural 3d portraits," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 20 364–20 373.

[7] G. Gafni, J. Thies, M. Zollhöfer, and M. 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)*, June 2021, pp. 8649–8658.

[8] K. Kania, K. M. Yi, M. Kowalski, T. Trzciński, and A. Tagliasacchi, "Conerf: Controllable neural radiance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 18 623–18 632.

[9] J. Sun, X. Wang, Y. Zhang, X. Li, Q. Zhang, Y. Liu, and J. Wang, "Fenerf: Face editing in neural radiance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 7672–7682.

[10] Y. Ren, G. Li, Y. Chen, T. H. Li, and S. Liu, "Pirenderer: Controllable portrait image generation via semantic neural rendering," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 13 739–13 748.

[11] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. 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.

[12] H. Kim, P. Garrido, A. Tewari, W. Xu, J. Thies, M. Niessner, P. Pérez, C. Richardt, M. Zollhöfer, and C. Theobalt, "Deep video portraits," *ACM Trans. Graph.*, vol. 37, no. 4, jul 2018. [Online]. Available: https://doi.org/10.1145/3197517.3201283

[13] Y. Deng, J. Yang, D. Chen, F. Wen, and X. 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.

[14] J. Thies, M. Zollhöfer, M. Nießner, L. Valgaerts, M. Stamminger, and C. Theobalt, "Real-time expression transfer for facial reenactment," *ACM Trans. Graph.*, vol. 34, no. 6, oct 2015. [Online]. Available: https://doi.org/10.1145/2816795.2818056

[15] P. Garrido, L. Valgaerts, H. Sarmadi, I. Steiner, K. Varanasi, P. Pérez, and C. Theobalt, "Vdub: Modifying face video of actors for plausible visual alignment to a dubbed audio track," *Comput. Graph. Forum*, vol. 34, no. 2, p. 193–204, may 2015. [Online]. Available: https://doi.org/10.1111/cgf.12552

[16] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3d faces," in *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '99. USA: ACM Press/Addison-Wesley Publishing Co., 1999, p. 187–194. [Online]. Available: https://doi.org/10.1145/311535.311556

[17] T. Li, T. Bolkart, M. J. Black, H. Li, and J. Romero, "Learning a model of facial shape and expression from 4D scans," *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, vol. 36, no. 6, pp. 194:1–194:17, 2017. [Online]. Available: https://doi.org/10.1145/3130800.3130813

[18] Y. Feng, H. Feng, M. J. Black, and T. Bolkart, "Learning an animatable detailed 3d face model from in-the-wild images," *ACM Trans. Graph.*, vol. 40, no. 4, jul 2021. [Online]. Available: https://doi.org/10.1145/3450626.3459936

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

[20] A. Ranjan, T. Bolkart, S. Sanyal, and M. J. Black, "Generating 3d faces using convolutional mesh autoencoders," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[21] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, p. 139–144, oct 2020. [Online]. Available: https://doi.org/10.1145/3422622

[22] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014. [Online]. Available: https://arxiv.org/abs/1411.1784

[23] M. Kowalski, S. J. Garbin, V. Estellers, T. Baltrušaitis, M. Johnson, and J. Shotton, "Config: Controllable neural face image generation," in *European Conference on Computer Vision (ECCV)*, 2020.

[24] Z. Shu, E. Yumer, S. Hadap, K. Sunkavalli, E. Shechtman, and D. Samaras, "Neural face editing with intrinsic image disentangling," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5444–5453.

[25] P. Hedman, P. P. Srinivasan, B. Mildenhall, J. T. Barron, and P. Debevec, "Baking neural radiance fields for real-time view synthesis," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 5875–5884.

[26] A. Pumarola, E. Corona, G. Pons-Moll, and F. Moreno-Noguer, "D-nerf: Neural radiance fields for dynamic scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 10 318–10 327.

[27] K. Park, U. Sinha, J. T. Barron, S. Bouaziz, D. B. Goldman, S. M. Seitz, and R. Martin-Brualla, "Nerfies: Deformable neural radiance fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 5865–5874.

[28] K. Park, U. Sinha, P. Hedman, J. T. Barron, S. Bouaziz, D. B. Goldman, R. Martin-Brualla, and S. M. Seitz, "Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields," *ACM Trans. Graph.*, vol. 40, no. 6, dec 2021.

[29] E. Tretschk, A. Tewari, V. Golyanik, M. Zollhöfer, C. Lassner, and C. 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)*, October 2021, pp. 12 959–12 970.

[30] Z. Li, S. Niklaus, N. Snavely, and O. 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)*, June 2021, pp. 6498–6508.

[31] W. Xian, J.-B. Huang, J. Kopf, and C. Kim, "Space-time neural irradiance fields for free-viewpoint video," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 9421–9431.

[32] C. Gao, A. Saraf, J. Kopf, and J.-B. Huang, "Dynamic view synthesis from dynamic monocular video," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 5712–5721.

[33] S. Athar, Z. Shu, and D. Samaras, "Flame-in-nerf: Neural control of radiance fields for free view face animation," 2021. [Online]. Available: https://arxiv.org/abs/2108.04913

[34] L. Liu, J. Gu, K. Z. Lin, T.-S. Chua, and C. Theobalt, "Neural sparse voxel fields," *NeurIPS*, 2020.

[35] C. Sun, M. Sun, and H.-T. 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)*, June 2022, pp. 5459–5469.

[36] L. Wu, J. Y. Lee, A. Bhattad, Y. Wang, and D. Forsyth, "Diver: Real-time and accurate neural radiance fields with deterministic integration for volume rendering," 2021.

[37] T. Hu, S. Liu, Y. Chen, T. Shen, and J. Jia, "Efficientnerf efficient neural radiance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 12 902–12 911.

[38] Q. Xu, Z. Xu, J. Philip, S. Bi, Z. Shu, K. Sunkavalli, and U. Neumann, "Point-nerf: Point-based neural radiance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 5438–5448.

[39] T. Neff, P. Stadlbauer, M. Parger, A. Kurz, J. H. Mueller, C. R. A. Chaitanya, A. Kaplanyan, and M. Steinberger, "DONeRF: Towards real-time rendering of compact neural radiance fields using depth oracle networks," *Computer Graphics Forum*, vol. 40, no. 4, pp. 45–59, jul 2021. [Online]. Available: https://doi.org/10.1111/cgf.14340

[40] A. Kurz, T. Neff, Z. Lv, M. Zollhöfer, and M. Steinberger, "Adanerf: Adaptive sampling for real-time rendering of neural radiance fields," 2022.

[41] Y. Hong, B. Peng, H. Xiao, L. Liu, and J. Zhang, "Headnerf: A real-time nerf-based parametric head model," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 20 374–20 384.

[42] Y. Guo, K. Chen, S. Liang, Y. Liu, H. Bao, and J. Zhang, "Ad-nerf: Audio driven neural radiance fields for talking head synthesis," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[43] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen, J. Chen, J. Chen, Z. Chen, M. Chrzanowski, A. Coates, G. Diamos, K. Ding, N. Du, E. Elsen, J. Engel, W. Fang, L. Fan, C. Fougner, L. Gao, C. Gong, A. Hannun, T. Han, L. V. Johannes, B. Jiang, C. Ju, B. Jun, P. LeGresley, L. Lin, J. Liu, Y. Liu, W. Li, X. Li, D. Ma, S. Narang, A. Ng, S. Ozair, Y. Peng, R. Prenger, S. Qian, Z. Quan, J. Raiman, V. Rao, S. Satheesh, D. Seetapun, S. Sengupta, K. Srinet, A. Sriram, H. Tang, L. Tang, C. Wang, J. Wang, K. Wang, Y. Wang, Z. Wang, Z. Wang, S. Wu, L. Wei, B. Xiao, W. Xie, Y. Xie, D. Yogatama, B. Yuan, J. Zhan, and Z. Zhu, "Deep speech 2: End-to-end speech recognition in english and mandarin," in *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ser. ICML'16. JMLR.org, 2016, p. 173–182.

[44] S. Sharma and V. Kumar, "Voxel-based 3d face reconstruction and its application to face recognition using sequential deep learning." *Multimed Tools Appl*, vol. 79, p. 17303–17330, 2020.

[45] J. 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*, vol. 3, 2000, pp. 314–317 vol.3.

[46] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[47] B. Deng, J. T. Barron, and P. P. Srinivasan, "JaxNeRF: an efficient JAX implementation of NeRF," 2020. [Online]. Available: https://github.com/google-research/google-research/tree/master/jaxnerf

[48] S. Sanyal, T. Bolkart, H. Feng, and M. Black, "Learning to regress 3d face shape and expression from an image without 3d supervision," in *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019.

[49] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[50] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *CVPR*, 2018.

[51] T. Akenine-Möller and B. Johnsson, "Performance per what?" *Journal of Computer Graphics Techniques*, vol. 7.

APPENDIX

Task	Semester	Semester 2, 2022										Summer Break		Semester 1, 2023											
	Week	1-7	8	9	10	11	12	13	14	15	16	21/11 - 23/12	2/1 - 26/2	1	2	3	4	5	6	7	8	9	10	11	12	13
Data collection and processing																										
Literature review and proposal writing																										
Implementation of RigNeRF																										
Oral Preseantation preparation																										
Oral Preseantation																										
Baseline models experiments on collected dataset																										
DECA implementation and experiments																										
MICA implementation and experiments																										
Voxel field (Density) implementation and experiments																										
Voxel field (Feature vector) implementation and experiments																										
Models evaluation and error analysis																										
Models adjustment and finetune																										
Thesis writing																										Submission

Figure 6: Landscape Timeline.