On the Democratization of Realistic 3D Head Avatar Generation and Reconstruction

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science
in
Electronics and Communication Engineering by Research

by

Pranav Manu
2020112019
pranav.m@research.iiit.ac.in

International Institute of Information Technology
Hyderabad - 500032, INDIA
June, 2025

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled ***"On the Democratization of Realistic 3D Head Avatar Generation and Reconstruction"*** by Pranav Manu, has been carried out under the supervision of Dr. Avinash Sharma and Prof. PJ Narayanan and has not been submitted elsewhere for a degree.

18.06.2025		
Date	Advisor 2:	Advisor 1:
	Prof. PJ Narayanan	Dr. Avinash Sharma

To those who...

... stand unshaken by the indifference of the universe.

Acknowledgments

This thesis is the product of the collective efforts and continuous support from my advisors, family and friends.

My sincere respect and profound gratitude are owed to my advisor, Dr. Avinash Sharma. Without his guidance, I might not have embarked on a research career or achieved my current level of involvement. His unwavering belief in me was a crucial source of reassurance, and he taught me invaluable lessons on navigating rejection and failure. His stories often provided timely wisdom during moments of stress or anxiety. Dr. Sharma significantly expanded my opportunities and exposure while granting me the freedom to explore my interests. He fostered an open and discussion-oriented lab environment, a legacy his students continue to uphold. He has always been a pillar of support for me, someone whom I can trust and look up to.

Moreover, I would also like to thank Dr. PJ Narayanan, my co-advisor, who supported me during my times of uncertainty and doubt. He allowed me freedom and trusted my judgment, which helped me gain confidence in my own work. His curiosity in various field encouraged me to remain curious as well.

I would like to thank my lab senior, a mentor and a friend, Astitva Srivastava, who patiently tolerated me throughout my thesis. We shared in the challenges and celebrated the successes of all my projects. His academic and personal support over the past three years has been immeasurable. I valued our ability to brainstorm, debate, and exchange ideas freely. I am truly grateful for all his efforts.

I also wish to thank Amogh Tiwari, one of my first acquaintances at CVIT. He provided exceptional help during my initial research project and consistently encouraged me towards greater discipline. I am grateful for the mentorship of my senior, Shanthika Naik, in geometry processing, who became a wonderful friend for late-night conversations. I also thank my seniors, Ishaan Shah, Rahul Goel, and Chandradeep, for their assistance with various technical questions throughout my projects.

My thanks also go to my friends from my lab batch. Collaborating with Aparna on course projects and beyond was a source of inspiration, and our discussions opened my eyes to the field of Sign Language generation. I am also grateful for the enjoyable times and insightful conversations shared with Maulesh, Adhiraj, Ayan, Deepti, and Chirag Parikh.

Furthermore, I thank my friends from various labs for their role in my personal development. Anjali's visits always brought joy to the lab. Chirag shared his knowledge of the human brain and introduced me to veganism. My ambitious projects with Sehgal were always fun, regardless of their outcome. Tanmay

was my reliable partner for any spontaneous plans or adventures. Rohan, my roommate and first friend at IIIT, was instrumental in making me a more outgoing person and even played a part in my joining CVIT.

Finally, I express my deepest and most profound gratitude to my parents and sister. Their unwavering support for my aspirations and their encouragement to pursue my dreams, even when they seemed uncertain, has been the foundation of my journey and the completion of this thesis. Their efforts have brought me to this point.

Abstract

The need for photorealistic head avatars has risen in the past decades, owing to the rising interest in the AR/VR media formats. An accurate representation of the head will be required in the near future, which is essential to facilitate communication between users, essentially enabling telepresence. The need for an improved in-person form of remote communication was made more clear during the recent COVID-19 pandemic and the ensuing lockdown, where millions of people had to stay away from their families and workplace for an extensive period of time. Besides, realistic facial avatars have proved immensely helpful in the movie and gaming industry, where they have often been used to either modify the actors' appearance itself, or to drive an entirely virtual but realistically looking digital character, depending on the demands of the narrative.

Capturing and reconstructing a realistic-looking head-avatar is not trivial, and requires an expensive setup of multiple synced cameras and lights, and a mathematical understanding of how light interacts with the skin, hair, cornea, etc. The capture of each subject is laborious and time-consuming. The creation of digital faces that are indistinguishable from real ones is a formidable challenge due to the "uncanny valley" phenomenon, where even minor deviations from realistic appearance can render a digital face unsettling to human observers. However, to achieve the applications of realistic head avatars in telepresence and AR/VR, the capture and thus creation of realistic digital replicas must be made accessible. Therefore, a need has arisen to search for methods that can reconstruct and create digital replicas that are photorealistic but also cheap. Our thesis aims to tackle this problem statement in two ways, one from the perspective of digital replica generation and the other from the perspective of creating a digital replica through reconstruction.

Our initial approach to make the creation of digital replicas efficient is a *textured head generation method conditioned on a descriptive text*. We aim to create a method that can generate a realistic-looking head avatar from a text description in an efficient manner, without requiring the manual intervention of artists or the use of highly specialised software like Blender or Maya. Therefore, it can generate textured head assets within seconds. However, the texture-based synthesis approach suffered from reduced realism because of the effects of baked-in lighting. Therefore, an approach is required that could construct a head avatar along with accurate material properties, such that it can be placed in any environment.

Our following work proposes a method to reconstruct a relightable head avatar using just a smartphone. Existing works that focus on creating a digital replica from a smartphone do not create a relightable head avatar or require a large amount of data prior to generating a head avatar with appropriate

material properties. Our proposed method introduces a novel capture strategy that can act as a small-scale replacement of lightstage-based capture, while only utilising a smartphone. We also propose an efficient head representation which allows real-time rendering and relighting, making it suitable for the application of telepresence.

We evaluate our proposed methods on public datasets, and conclude that our methods outperform state-of-the-art methods, both in the realms of text-based generation and capture-based reconstruction. We also release our own dataset, recorded using our novel capture setup. Additionally, we discuss our methods' limitations, offering insights into potential improvements and outlining promising future research directions. We hope that this thesis will substantially impact the field and accelerate progress in 3D Head generation and reconstruction.

Contents

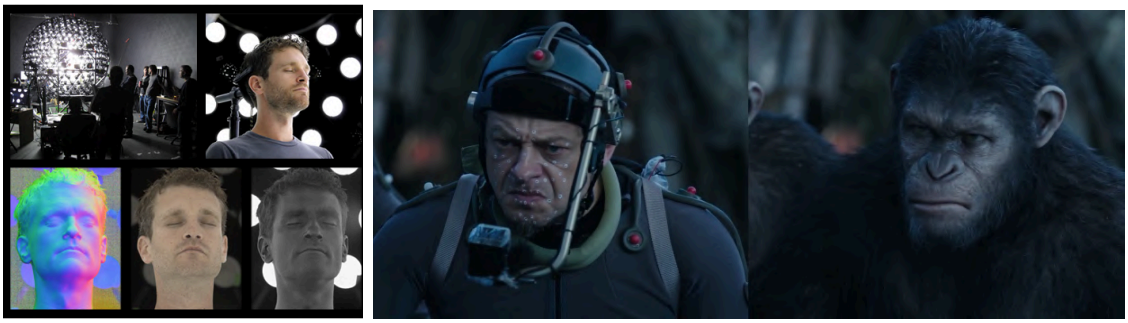
List of Figures

List of Tables

Chapter 1

Introduction

The general public media and forms of communication have seen a shift over the past centuries. Initial modes of media and communication were limited to text or writing, which lacked the subtleties of communication while people are talking in person. In the coming decades, and with advancing technology, a shift was seen from the written form of communication to an audio-based format. Audio is better at capturing nuances of expression and is able to capture the speaking style of the person, though it still lacks facial features or expressions of the person on the other end. With the advent of audio-visual communication, all of these problems seemed to have been solved; however, in the 21st century, the requirements of communication and media formats have evolved. Holoportation and telepresence are no longer limited to the spheres of science fiction and are actively being pursued by several giants like Meta and Microsoft. The idea is whether a person's presence can be placed in an environment where they are not present. A person's presence cannot be merely represented by the two spatial dimensions that generic forms of audio-visual media provide. Humans being three-dimensional forms of life, a third spatial dimension is required to fully and accurately represent a person in any environment. This is where 3D computer vision comes in. The major challenges lie in capturing the appearance of the person and then representing it accurately at the receiver's end in a photo-realistic manner, specifically for the face. Humans, being a social animal, have evolved to develop a keen sense of faces, to recognise social facial cues of other humans. Therefore, if a digital face deviates even slightly from how an actual face looks, it starts looking unreal or unnatural, which is usually called the 'uncanny valley'. Therefore, making or capturing realistic-looking faces for the purpose of video-games, telepresence, AR/VR is an active research area in the field of 3D computer vision. The presented thesis also aims to contribute towards this effort.



Light-stage capture setup to capture facial surface properties

Motion Capture used in movies to drive the expression of virtual characters

Figure 1.1 Applications of head capture

1.1 Digital Replica of Human head

1.1.1 Motivation

Photoreal faces have long been a requirement in the VFX industry. It is often required to de-age the actor, change the appearance, or replace the face of a stuntman with that of the actor. Although these goals can be achieved to a certain extent while shooting via makeup artists, the innovations in 3D computer graphics have made it possible to make these changes in the editing room, avoiding a lot of arduous facial makeup and reduced the prep time for shooting a scene. Multi-million dollar movies like Avatar use techniques derived from computer graphics and computer vision to capture the expression of a source actor and transfer it to a generated, realistic-looking avatar.

Besides movies such as *Irishmen*, *The Lord of the Rings*, facial expression capture and transfer have long been used in the gaming industry, which is bigger than the music and movie industry combined. Making realistic-looking digital replicas of actors is often a goal of big-budget video games. The digital replica of the actor must capture all the subtle expressions and appearance of the source actor, to effectively convey the emotion of the character across the screen. While foreground characters may require high fidelity details, the non-playable characters often can be of lower quality, but require unique identities. For this, graphic artists have to manually create a large number of faces and human assets. Therefore, a system that could quickly generate these assets has been desirable to the gaming industry. As mentioned earlier, telepresence has been a long-standing challenge, where the creation of a photorealistic digital replica is a crucial step for immersion and believability. Especially for communications, a realistic digital head is a must.

Existing approaches to creating a realistic head avatar have focused on capture using a light stage based [29] setup as seen in Figure 1.1, where the actor is seated within a dome of cameras and RGB lights. For a given timestep, the same facial expression is visible from multiple directions and can be viewed under different illumination conditions. The interaction of skin under various light conditions and viewing directions enables the estimation of properties of the surface of the face through an

analysis-by-synthesis approach. The resulting digital replica is photo-realistic, and can be placed in any environment realistically, because the interaction of the skin with environmental lighting can be mathematically calculated using the light transfer equation [36] and one of the several skin models [17, 35] presented in literature.

A lightstage capture is able to achieve very high-quality digital replicas, but setting up a lightstage is very expensive, owing to the hardware required for capturing, syncing and processing such a large amount of data in parallel. A more accessible approach would be to use commodity smartphone cameras to capture a realistic avatar as in [1, 47, 95]. However, without knowing the appearance of the skin under various illumination conditions and from various angles, it becomes an ambiguous problem to estimate the appearance of the head under a novel illumination setup. With companies investing billions into AR/VR experiences like the metaverse [77], capturing and rendering realistic avatars must be made accessible to open up the market to average users. The work done as a part of this thesis aims to democratise avatar creation from the aspect of both capture and text-based generation.

1.1.2 Challenges

Making a digital replica of a face becomes extremely difficult because the human mind is extremely perceptive to changes in facial features, and slight deviations from the actual appearance may cause it to look unnatural or uncanny. Therefore, the replica must appear photo-realistic. Capture and recreation of a photo-real head avatar has several challenges

- The most accurate way to capture a human head is using a light stage setup, which is neither scalable nor accessible to the average user. It requires an expensive setup, meticulous calibration and computationally intensive post-processing.

- Skin as a material is very hard to model, because of the layered nature of the skin. Not all light is reflected at the surface of the skin. Some light is initially reflected at the oily surface of the skin, while some goes through it, undergoes some scattering within the upper layers of the epidermis of the skin, and then is reflected back into the air. This makes modelling the skin mathematically very difficult. A raytracing-based approach, where each layer of the skin is modelled individually, would be the most accurate at the cost of being extremely computationally intensive. Several models for skin exist like [35] which present an approximate solution to the interaction of light with the skin.

- Besides skin, a human head consists of several other materials, such as those of the facial and head hair, teeth, tongue and eyes. All of these elements interact differently with light and need to be modelled accurately for a realistic-looking head avatar. Hair strands need to be modelled individually for them to be physically represented in a 3d environment, which is also computationally expensive.

- In the case of artificially generated human characters, the digital head must be modelled manually by an artist using 3D modelling and rendering software like Blender[9], Cinema4D[49], Maya[4]. The process of modelling is very strenuous and requires a strong familiarity with software.

- Editing of the head avatar also requires manual intervention by artists, which in turn, also requires an expertise in 3D modelling software as those mentioned above.

- There exist few to no public datasets consisting of multiple views of the head under different illuminations, which makes it difficult to train deep learning based methods to make generalizable, relightable head avatar models.

1.2 Accessible 3D head avatar generation and capture.

The problem of digital head avatar creation can be tackled from two aspects, one from the aspect of ease of generation by describing the asset to generate, and the other from the aspect of capturing a photoreal head avatar without relying on an expensive capture setup. The thesis presents approaches to both these aspects, which may help democratise the creation of digital head replicas in the near future.

1.2.1 Problem Statement

The thesis aims to tackle some of the challenges that exist in creating a realistic head capture without requiring an expensive light stage setup or long and tedious efforts by artists. The thesis aims to aid the artists and VFX engineers in speeding up the capture and creation of a photo-real digital avatar. Therefore, as suggested earlier, the thesis will target the problem from the aspect of the generation of digital head assets, given a simple description, as well as provide a novel procedure to capture a relightable digital head avatar without the use of a light stage setup. We leverage recently developed 2D latent diffusion models[66] for generating text-conditioned 3d assets. Generative diffusion models have transformed the field of 2D image synthesis, though it has not been fully explored in the realm of 3D, especially for human heads. We will explain in the upcoming sections that using 2D latent diffusion models for 3D generation is not straightforward, requiring a computationally intensive iterative optimisation step, which presents itself as a bottleneck in any asset generation pipeline.

With the goal of making the capture process accessible, we choose a commodity monocular smartphone camera as our device of choice. However, it is difficult to estimate how skin interacts with light under various conditions from a single camera view. A generic approach to capture may lead to a realistic head avatar, but the head avatar would not appear realistic when placed in a different environment. Our novel procedure for relightable head capture exploits how the skin preserves the polarisation of light, depending on which layer of the skin light has been reflected from.

1.2.2 Research Landscape

The two approaches mentioned above can broadly be divided into the following:

1. **Geneartive Approach:** The success of large-scale vision-language models like CLIP [2] has spurred significant interest in text-guided generation across various modalities, including 3D. To leverage these 2D vision-language models for 3D, several approaches like Score Distillation Sampling [58] and Score Jacobian Chaining [78]. These methods allow great diversity in generation, and allow text-conditioned generation without training on any annotated 3D dataset. However, this generic text to 3d generative models often suffer from the '*Janus Problem*' as mentioned in DreamFusion [58], where the 3d head model exhibits duplicated canonical features like double nose, 3 eyes, etc., on different sides, lacking a coherent and singular 3d structure. To overcome this, SDS was further extended to a setting specific to human heads, using a parametric head model like FLAME [44] in works like ClipFace[74] and [89]. Grounding the generative optimisation steps on a parametric model helped avoid the *Janus problem*, but they still suffered from the inherent bottleneck of SDS-based methods, which was due to the iterative optimisation-based approach, which required inference time optimisation for each textual condition.

2. **Capture Based methods:** The Gold standard for capture-based head avatar reconstruction is a lightstage, pioneered by Paul Debevec in [17], but it requires specialised hardware and calibration, and is compute-intensive. Alternate approaches have been developed that optimize an implicit representation like NeRF[51] as in INSTA[1]. The facial movements are represented by a learning deformation field conditioned on a parametric head model like FLAME [44]. These methods are computationally intensive and struggle to perform real-time rendering due to the volumetric ray marching approach used for rendering, where several points along a ray must be queried to get a final colour at a pixel. Later methods shifted to using a 3DGS[39] based explicit representation, which was guided by the parametric head model. Initial works like GaussianAvatar[61] relied on multiple simultaneous cameras, whereas further work like InstantSplat [72] aimed at creating a photoreal head avatar from monocular video. These head avatars could be rendered in real-time, but suffer from their inability to be relit or edited.

   Another branch of work [5, 17, 64] tackles this aspect from the lens of physically based inverse rendering to achieve a head avatar that is relightable under various environments. These methods use approximations to discern the properties of the skin from limited information that might be available from multiview RGB images. Though relightable, these methods create a head model that is static in nature and lacks expression-dependent appearance details like wrinkles.

The above highlighted challenges outline the need for a generative method which is fast and efficient, one which can create a diverse set of head avatars conditioned on a description, in a short time, while on the other end, a capture method which can combine the fidelity of a capture setup, while being as accessible to a user a phone camera. The presented thesis will address these issues.

1.3 Main Contributions of the Thesis

1. **Text Conditioned Generation of 3d Head Avatars:** We propose a pipeline that can generate textured head avatars conditioned on text description in a single feed-forward setup, allowing a method to create diverse-looking head avatars with minimal text description in under a minute.

2. **DuoPolo: Polaroid Filter based Dual Capture of Head using a smartphone camera**: A novel capture setup which utilises a set of cheap polaroid filters to capture a dynamic head performance of an actor. The polaroid filter allows us to better disentangle diffuse and specular reflections off the skin.

3. **Relightable and Editable Head Avatar representation:** We propose a novel representation of head avatars based on primitives on a texture map, which allows shader-based physical rendering. This enables the head model to be relit in any environment. It can build a relightable head avatar from the data captured in the DuoPolo dataset.

1.4 Thesis Roadmap

In this chapter, we briefly discussed about 3D head avatar generation and reconstruction and the challenges it presents. Furthermore, we discussed briefly recent advancements in the field and the research gaps that still persist.

In *Chapter-2*, we will provide relevant background details, which would be crucial for understanding upcoming chapters, and the major contributions of the thesis. We will briefly discuss various representations of 3D geometry and parametric head models that have been used or built upon in the work done during this thesis.

Chapter-3 discusses the task of generation of a textured neural parametric head avatar conditioned on text, in a fast and efficient manner. We will elaborate on the method, and through our evaluations, we will demonstrate that our method outperforms existing methods both qualitatively and quantitatively.

In *Chapter-4* we will present a novel head representation based on 2D Gaussian splats and a novel capture setup which can disentangle the diffuse and specular appearance of the surface of the skin. We quantitatively define that our method outperforms other methods on the proposed dataset, while performing comparably to existing methods on other monocular datasets.

In *Chapter-5*, we propose an improvement to the head representation introduced in *Chapter-4*, making it more efficient in terms of computational expense, while also increasing its qualitative and quantitative performance, which makes our method better than all existing methods when tested across various datasets. We will discuss the improvements made over the previous approach.

Chapter 2

Background

Before proceeding with the thesis problem statement, we will give a brief overview of relevant topics and related literature, which will help build a framework for understanding the research problem. The following section discusses the various 3D representations in use today, their advantages and disadvantages, and our reasoning behind using some of these representations. Furthermore, we discuss few parametric head models [44, 75] which were utilized in this thesis. A brief introduction is also given to Conditional Generative models.

2.1 3D Representations

It is crucial to represent recorded 3D data digitally, and there exist several approaches to it, each with its own perks and drawbacks. In this section, we discuss briefly the most popular representations.

2.1.1 Explicit Representations

An explicit representation of a 3D surface defines each location on the surface explicitly or directly. Most common forms of 3D representations used in computer graphics fall under this category.

2.1.1.1 Point Cloud

A point cloud represents a scene or an object as an unordered set of discrete points in a coordinate system, usually the Cartesian coordinate system. Each point is defined by its position in a given coordinate system. Many scanners (e.g. LIDAR) for 3D sensing outputs their sensory data in point cloud format. Any point cloud can be converted to a triangle-based mesh, better for visual representation using Poisson Surface Reconstruction [38], Delaunay triangulation, etc.

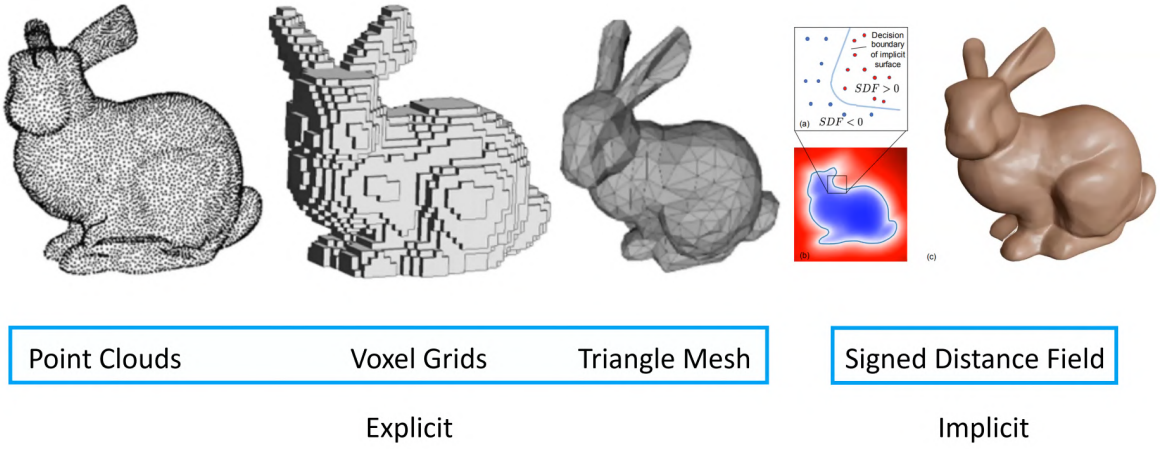


Figure 2.1 Different representations to represent a 3D surface. Source:[22]

2.1.1.2 Mesh

A mesh is defined by the position of the vertices, similar to a point cloud. However, unlike a point cloud, it also includes information about the connectivity of these vertices in the form of edges of a face, which can be a triangle, quad, etc. A face with N edges is defined by the indices of N vertices that constitute the face. The vertices of the mesh may also contain several attributes, such as:

- **Vertex normals**: These vectors, ideally perpendicular to the surface at the vertex, are fundamental for lighting calculations, determining how light reflects off the surface.
- **Vertex texture coordinate**: These 2D coordinates map a vertex to a specific point on a 2D texture image, enabling the application of image textures to the mesh surface.
- **Vertex Tangents (and Bitangents)**: Often used in conjunction with normals and texture coordinates, tangents define a local coordinate frame at the vertex, which is crucial for advanced shading techniques like normal mapping.
- **Weights (Used for animation)**: Primarily used in skeletal animation, these values define the influence of one or more "bones" or joints on a vertex's position during deformation.

2.1.1.3 Gaussian Splats

3d Gaussian splitting [39] is a highly efficient technique of scene representation and novel view synthesis in which the scene is represented as a set of Gaussian primitives, each associated with its attributes. The Gaussian primitives are splatted onto the image plane in a differentiable manner to render the scene. Each Gaussian primitive is defined by the following attributes:

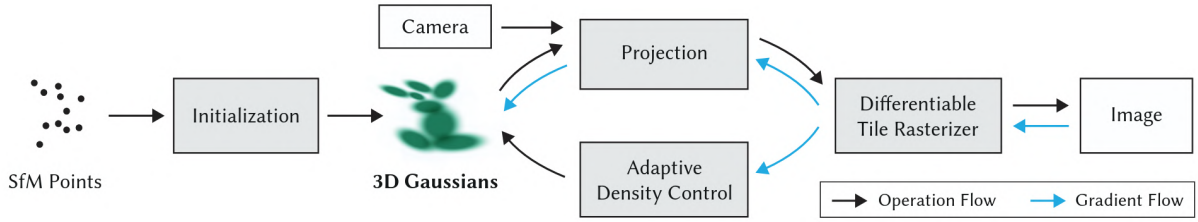


Figure 2.2 Optimization of 3D Gaussian splats. Source:[39]

- **Position** (μ): Defines the position of the centre of the primitive in 3D space. It represents the mean of the Gaussian, where the contribution of the Gaussian to the appearance is maximum.
- **Covariance** (Σ): Defines the size and orientation of the gaussian primititive in 3D space. It can be parameterised using two factors, scale (**S**) and rotation (**R**) of the Gaussian primitive.

$$\Sigma = RSS^T R^T$$

- **Opacity** (α): Defines the contribution of the gaussian to the appearance of the scene. It is a measure of the transparency of the primitive.
- **Color**: Defines the appearance of each gaussian primitive. There can be several approaches to define the appearance of the gaussian. A naive approach would be to associate an RGB value to each of the gaussian. The drawback of such an approach is that the gaussian primitive would not be able to represent view dependent appearance changes, such as reflections, and would appear the same from all directions. To mitigate this, the appearance is represented using spherical harmonics coefficients. Spherical Harmonics are a set of orthogonal functions which can be used to define any function defined on a unit sphere. View dependent appearance can be considered as a function which maps the viewing angle to a colour value:

$$f : (\theta, \phi) \rightarrow \mathbb{R}^3$$

A Gaussian in 3D space is defined as:

$$G(x) = \exp\left(-\frac{1}{2}x^\top \Sigma^{-1} x\right)$$

The 3d Gaussian can be projected onto the 2D image plane based on the camera parameters. The projected 2D gaussian is splatted on the image grid via rasterization. The contribution of all the gaussian at a pixel is computed via alpha compositing, and a final pixel color is derived.

3DGS based representation allows realtime rendering of photorealistic scenes, owing to the view dependent appearance of each gaussian primitive, and an efficient rasterization step.

2.1.2 Implicit representation

A surface can be defined as the zero-level set of a scalar-valued function, which is known as the implicit representation of the surface.

$$F(x, y, z) = 0$$

2.1.2.1 Signed Distance Field

A Signed Distance Field [21, 80] or a Signed Distance Function F maps a point in a Euclidean space to its orthogonal distance from the surface of a geometric shape. The function has positive values inside the geometric shape and negative values outside the geometric shape. This convention might be reversed in some cases.

$$F : R^3 \rightarrow [-1, 1]$$

2.1.2.2 Unsigned Distance Field

Similar to a signed distance field, each point in 3d space is mapped to a value which represents the orthogonal distance of the point to the surface of the geometric shape, but unlike a signed distance field, all points have a non-negative value. This convention is useful for surfaces that might not be closed and do not have a definite boundary that demarcates the interior and exterior regions. Surfaces like those of garments are open surfaces that can be effectively defined using an unsigned distance field. [15]

$$F : R^3 \rightarrow [0, 1]$$

2.2 3D Appearance

Several approaches have been developed to define the appearance of 3D geometry. We briefly discuss some of the commonly used methods that are employed.

2.2.1 Per Vertex Color

A naive method for defining the appearance of the mesh is to give a colour attribute to each vertex of the polygonal mesh, and interpolate the colour values for each point inside the polygon. The appearance of the mesh can therefore be saved as another vertex attribute of the mesh. However, appearance resolution is limited by the number of vertices of the mesh. As a result, to represent high-frequency appearance details, a large number of vertices are required, making this representation very inefficient.

2.2.2 Textured Appearance

A mesh can be parameterised to map the 3D surface onto a 2D plane. The mapping of the 3d surface on the 2D plane is defined by a vertex attribute, called per-vertex uv coordinate, which defines where

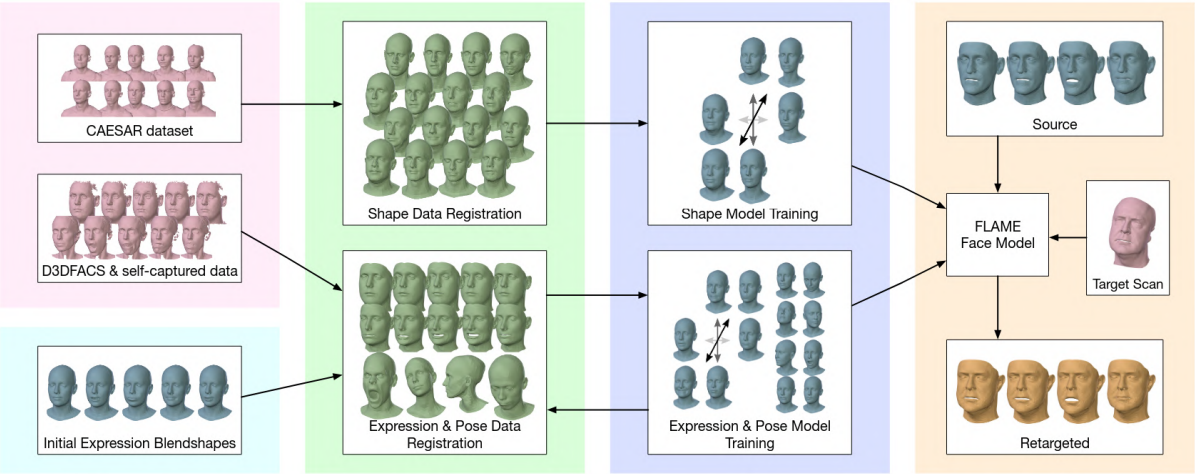


Figure 2.3 Overview of FLAME face registration, training, and expression transfer. Source:[44]

a vertex of a polygonal mesh maps to in a normalised 2D coordinate map. The mapping of each point on the triangle can be computed by interpolating the uv coordinates of the vertex. There may be several mappings possible, but the mapping with the least warping or distortion is preferred. The distortions can quantitatively be evaluated by computing the change in area of the triangles, and as a result, by the change in angles of the polygon in 3D space, and after mapping to the 2D plane. Several approaches [73, 20] have been explored for UV parameterization of meshes. The mapping of 3D surface allows each point to be mapped to a 2D plane, and therefore each point on the mesh surface can be associated with a 2D image, through a process called texture mapping. Texture mapping allows us to use an arbitrary resolution appearance for any mesh. The resolution of the appearance/texture becomes independent of the resolution of the mesh. Texture-based appearance allows efficient storage for high-resolution features while allowing greater editing control to the user or artist.

2.3 FLAME: Parametric Head Model

FLAME (Face Learned with an Articulated Model and Expression) [44] is a parametric 3D statistical head model which can represent a face with a set of shape ($\beta \in \mathbb{R}^{300}$) parameters, expression ($\psi \in \mathbb{R}^{100}$) parameters and pose ($\theta$) parameters. It uses vertex-based linear blend skinning (LBS) with corrective blend shapes. The flame head model consists of a template mesh with a number of vertices ($N = 5023$) and ($K = 4$) joints for neck, jaw and 2 eyeballs. The pose parameters were learned from the data captured by the authors, see Figure 2.3. The expression model uses the data captured by the authors and the D3DFACS[56] data. Since the Flame head model utilises a template mesh that can be deformed to represent a diverse set of face shapes and expressions, a common UV map can be used for all the head

models, making it possible to apply the same texture map to a diverse set of faces. The parametric head model is defined by the following parameters:

- **Shape Parameters (β):** $B_S(\vec{\beta}; \mathcal{S}) : \mathbb{R}^{|\vec{\beta}|} \rightarrow \mathbb{R}^{3N}$ These are the coefficients that represent the weight of each orthonormal shape basis. When weighed and added together, they can represent the geometry of a face in neutral expression. The shape space, or the shape basis vectors, is learnt via Principal Component Analysis from the CAESAR [65] body scan dataset.

- **Expression Parameters (ψ):** The expression blendshapes can be considered as a mapping from expression parameters to vertex offsets as $B_S(\vec{\psi}; \mathcal{E}) : \mathbb{R}^{|\vec{\psi}|} \rightarrow \mathbb{R}^{3N}$. These parameters can represent a wide range of expressions which are independent of the shape of the face. Similar to shape, expression blendshapes are also a linear combination of orthnormal expression basis vectors.

- **Pose Parameters (θ):** The rigid transformations of the template mesh, like head rotation, translation, eye pose and jaw pose, are represented by these parameters. The pose blendshapes can be considered as a mapping from pose parameters to K rotation matrices $R(\vec{\theta}) : \mathbb{R}^{|\vec{\theta}|} \rightarrow \mathbb{R}^{9K}$

The shape blendshapes $B_S(\vec{\beta}; \mathcal{S})$, pose blendshapes $B_P(\vec{\beta}; \mathcal{P})$ and expression blendshapes $B_E(\vec{\psi}; \mathcal{E})$ can be used to deform a template mesh $\bar{T} \in \mathbb{R}^{3N}$.

$$M(\vec{\beta}, \vec{\theta}, \vec{\psi}) = W(T_P(\vec{\beta}, \vec{\theta}, \vec{\psi}), J(\vec{\beta}), \vec{\theta}, \mathcal{W}),$$

where T_P is the template mesh with pose, expression and shape offsets applied.

$$T_P(\vec{\beta}, \vec{\theta}, \vec{\psi}) = \overline{T} + B_S(\vec{\beta}; \mathcal{S}) + B_P(\vec{\theta}; \mathcal{P}) + B_E(\vec{\psi}; \mathcal{E})$$

Chapter 3

Text-Guided Generation of Textured Neural Parametric Head Avatars

In this chapter, we discuss our proposed approach for text-driven 3D textured head generation, which can accept simple natural language text prompts describing the appearance and shape of the head, and our model, ClipHead, generates a neural parametric head model with accurate geometry and high-quality texture maps. In contrast to existing methods that rely on conventional parametric head models like FLAME [44], our approach leveraged Neural Parametric Head Model (NPHM) [26], which represents a head as a Signed Distance Field, with disjoint latent codes for shape, identity and expressions. To enable text-driven generation, we proposed two lightweight mapping networks trained in a weakly supervised manner, which align the embedding space of CLIP[2] with the identity and expression latent space of NPHM, which allows us to navigate NPHM's latent space with text. The network accepts CLIP's embedding via text encoding and predicts a latent code for expression as well as identity. When these latent codes are decoded via NPHM pre-trained network, we get a 3D head geometry. Given that NPHM does not support texture-based appearance, we proposed a novel two-stage technique for text-driven generation of texture maps. This approach leverages a recently introduced controllable diffusion model, which allows spatial control in text-driven image synthesis. Semantically consistent UV parameterization was performed for the generated head mesh to derive unwrapped surface normal maps. These surface normal maps steer the diffusion process, thereby generating realistic textures conditioned on a text prompt in a coarse-to-fine manner.

3.1 Introduction

Human faces are fundamental to communication and identity, so accurately representing them in digital spaces is key for creating believable and emotionally engaging experiences for users. In the contemporary realm of virtual media, 3d face and head modelling is of significant importance for a wide range of applications, such as gaming and mixed reality, where it enhances realism and immersion. It enables the creation of expressive and personalized avatars, contributing to character design and animation. In the medical field, 3d face modelling also supports patient-specific simulations and surgical planning. Moreover, it finds utility in film production, education, teleconferencing, and scien-

tific research. In medicine, 3d face modelling also supports patient-specific simulations and surgical planning.

Traditional methods for 3d head modelling and animation often entail time-consuming manual sculpting or complex 3d scanning techniques, limiting their scalability and diversity. Furthermore, these approaches struggle to capture the rich variability of facial expressions and identities required for realistic human-computer interactions. A prominent approach in tackling these challenges involves the utilization of parametric head models like FLAME [44] as explained earlier, which effectively capture shapes and expressions within a low-dimensional parametric space. These models have demonstrated remarkable success by accommodating sparse inputs, mitigating noise, and offering a concise 3D representation. However, their primary drawback is the inability to capture intricate local surface details and their reliance on a fixed topology template mesh, which restricts their representation of diverse hairstyles. Additionally, the parameters for shape and expression are often coupled, providing limited control over the generation.

The recently proposed ClipFace [74] leverages FLAME's parametric geometry and introduces a StyleGAN-ADA[37] based generative model to jointly train a texture mapper and an expression mapper on a given FLAME [44] face mesh with initial UV texture codes. To facilitate controllable editing, the authors use a pre-trained CLIP[2] model to guide expression and texture mapping based on a prompt. However, the method suffers from the inherent limitations of parametric head models, and there is no direct support for controlling the shape of the face simply via the input prompt itself. Moreover, the diversity in output, for both expression and texture, is limited by the initial texture codes, and the texture synthesis needs to be retrained for a different prompt. Another recent work [89] introduced Describe3D, which is a 3D-face-text dataset containing 3DMM parameters for 1,627 3D face models and corresponding fine-grained, manually-labelled facial features, which include 25 facial attributes and 3 to 8 options describing the facial feature for each attribute. The authors then propose a text parser network that takes a CLIP encoding of a text prompt and predicts a descriptive code, trained in a supervised fashion on the Describe3D dataset. During testing, optimization is performed on concrete and abstract synthesis pipelines using CLIP-loss and differentiable rendering loss to convert the descriptive code into predicted target 3DMM parameters, along with a UV texture map. However, the method requires strong supervision in the form of manual text annotations, and the shape variation is limited to just 1,627 face shapes. Additionally, the texture synthesis is also limited as there is no support for the level of text-guided texture editing similar to what ClipFace offered.

Neural parametric head models (NPHM), proposed in [26], represent human head geometry in a canonical space using an SDF, and morph the resulting geometry to a posed space using a forward deformation field, thereby disentangling the head geometry into two disjoint latent spaces: identity and expression. The neural disentanglement provides higher levels of detail and a more flexible 3D head generation capability by decoding sampled latent codes for identity and expression, unlike PCA-based low-dimensional parametric head models based on 3D mesh, which are not able to capture high-frequency geometric details. However, estimating latent code parameters for a specific target identity

and expression requires a point cloud representation of the target head, limiting the accessibility and controllability of generating 3d head geometries.

In this section, we will discuss the work presented in the paper ClipHead where we proposed a novel method for the controllable generation of 3D head meshes from natural language prompts, along with high-quality texture maps. We harnessed the capabilities of CLIP [2], an encoder architecture that can encode images and text into a common representation space, where words that describe the image are encoded close to the embedding of the image itself. To facilitate the text-guided generation of NPHM meshes with varied identities and expressions, we use two mapping networks—one for mapping the CLIP embedding vector to the identity latent code of NPHM, and the other for mapping the same CLIP embedding vector to the expression latent code. We train both networks (MLPS) in a weakly supervised fashion, without requiring pairwise 3d head and text description pairs, alleviating the need for manual and labourious text annotations. 3d head geometry is generated by conditioning the pre-trained NPHM decoder using the predicted latent codes to query Signed Distance Function (SDF) values at densely sampled points, which are subsequently used to obtain a 3d polygon mesh using marching cubes.

Owing to the unavailability of a large paired dataset of text and 3D head meshes, we leverage 2D latent diffusion models, introduced in [66], for generating diverse text-guided appearances for various head geometries. However, we require the text-driven synthesis to be in accordance with the intended appearance of a given 3D head geometry. Therefore, we propose to use ControlNet [93], which allows additional spatial conditioning along with the text to provide more control over the latent diffusion process. We used the unwrapped surface normal as spatial conditioning to generate a UV texture (RGB) map via the ControlNet model. For training the aforementioned variant of ControlNet (sec. subsection 3.3.3), we utilised the dataset introduced in NPHM [26], comprising 5,201 high-fidelity textured 3D scans of human heads. However, each 3D scan mesh exhibits varying and unstructured UV parametrization (see Fig. Figure 3.1). Semantically meaningful and aligned UV maps are required to provide a useful hint to ControlNet. Therefore, we developed a technique to reparametrize the 3D head meshes and roughly align their UV coordinates (sec. subsection 3.3.2), which facilitates the localization of semantically similar regions of all head meshes within the same UV space region (see Fig. Figure 3.2). During inference, we feed the aligned UV normal map of the given NPHM head mesh to the ControlNet and generate the UV texture map guided by a text prompt.

The above approach allowed us to avoid the optimization at the inference approach, which was prevalent earlier and led to a significant bottleneck in terms of speed. Our single feed-forward approach allowed us to generate diverse head geometries and appearances within a minute, while outperforming the state-of-the-art methods of the time, both qualitatively and quantitatively.

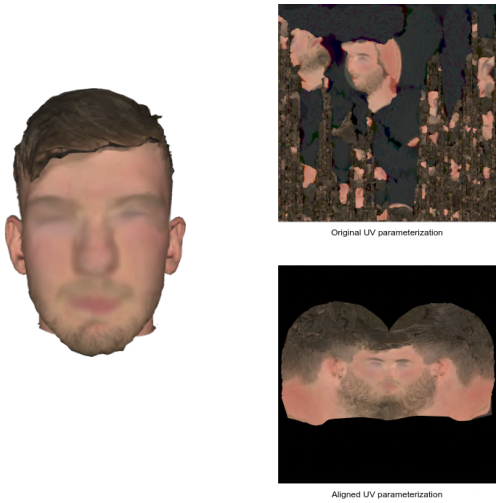


Figure 3.1 Re-parametrization of a sample mesh from NPHM dataset using our proposed technique to get aligned UV map with textures projected from original texture map, used for training our *Texture Synthesis* module ($ControlNet_{uv}$). Facial details have been blurred to protect identity.

3.2 Literature Review

3.2.1 Text-to-3D Optimization based approaches

After the success of Latent Diffusion models [66] in text conditioned image generation, a significant research effort started to focus on the challenging problem of text conditioned 3D generation. There exists no large annotated dataset for 3D assets, unlike those for 2D images, like LAION 5B [71], which made it possible to train text-conditioned latent diffusion models, utilising the clip embedding space. To overcome the unavailability of 3D annotated data, the concept of Score Distillation Sampling (SDS) was popularized by DreamFusion[58], in which an implicit volumetric representation like NeRF [51] is iteratively optimized. Camera views are randomly sampled in the scene, and differentiable rendered into 2D views. A pretrained latent diffusion model can then be used to estimate the gradient of the log-density that guides the 3D model to be consistent with the text prompt, in terms of embedding of the text and the rendered view of the model in clip embedding space. Though this approach leverages the diversity of 2D latent diffusion models, the iterative optimization step proves to be a bottleneck in asset generation due to its slow nature. Moreover, as discussed in DreamFusion [58], SDS-based approaches often suffer from the "Janus Problem", where certain features might be repeated on the surface, or mirrored along an axis. This may cause artefacts such as two noses, three eyes, two mouths, etc.

The issue of multi-faced Janus can be resolved if the optimisation-based generative approach is grounded by a parametric head model while generating head avatars. Therefore, works like ClipFace [74] enabled text-guided generation of head avatars by optimising the texture map of a FLAME-based model via differentiable rendering. It leverages an adversarial generative network for appearance syn-

thesis with CLIP [2], facilitating text-driven generation and editing. However, it inherits FLAME's geometric limitations and cannot generate details like different hairstyles, while also requiring test-time optimization for texture synthesis.

Another work HiFi-Face [89] provides fine-grained geometric manipulation by training a supervised text conditioned 3DMM parameter estimation network, guided by CLIP. The major limitation is the reliance on strong supervision in the form of highly descriptive text-prompts paired with 3DMM parameters, which are manually annotated. Furthermore, the diversity in appearance of the generated head model is severely limited due to the inherent limitations of the underlying 3DMM.

3.2.2 3D Morphable Models (3DMM)

3D Morphable Models are the most widely adopted method for representing 3D human faces pioneered by BFM[7] and more recently FLAME[44]. 3DMM are a form of statistical head models that represent shape and expressions in a low-dimensional parametric space, which is learnt from a large dataset of 3D head scans. The high-dimensional geometric information (high-dimensional due to a large number of vertices in the physical scan and three dimensions for each vertex) is pAlthough the parametric nature of 3DMM allows for an efficient representation, it also limits the expressiveness of the head model, as all head models end up having a fixed mesh topology.ad from sparse inputs, such as those from monocular images or from RGBD images. These parametric head models can even be animated by simply controlling the low-dimensional parametric spaces.

Although the parametric nature of 3dmm allows for an efficient representation, they also limit the expressiveness of the head model, as all head models end up having a fixed mesh topology. 3DMM-based approaches often fail to capture fine geometric details like wrinkles and hair strands due to the low-dimensional nature of the parametric space. They further fail to generalise to shapes and expressions far away from the training distribution, and cannot provide a lot of diversity in appearance.

3.2.3 Neural Parametric Head Models(NPHMs)

To address the limitations of traditional 3DMMs, recent research has explored neural network-based parametric models, which do not rely on a fixed topology mesh, but instead rely an implicit representation for the head model. NPHM [26] represents head geometry by using Multi Layer Perceptrons (MLPs) that decode latent codes into an SDF-based representation. A mesh can be extracted from the SDF by running marching cubes.

NPHM representation disentangles the head shape like bone structure, from the changes in geometry due to expressions by learning two disjoint latent spaces, one for identity and the other for expression, conditioned on expression latents. This allows for more granular and independent control over these attributes. An SDF-based representation also allows more complex facial features to be captured, like hair strands and wrinkles, to some extent.

Although a good parametric representation, a primary limitation of NPHM is that the generation of the parametric head model requires a point cloud input from RGBD cameras. The estimation of latent codes, therefore, required a slow, iterative optimisation process. This hindered direct control and generation. Furthermore, NPHM, being an SDF-based representation, is texture agnostic and focused only on geometric representation.

3.3 Proposed Text-driven 3D Textured Parametric Head Generation

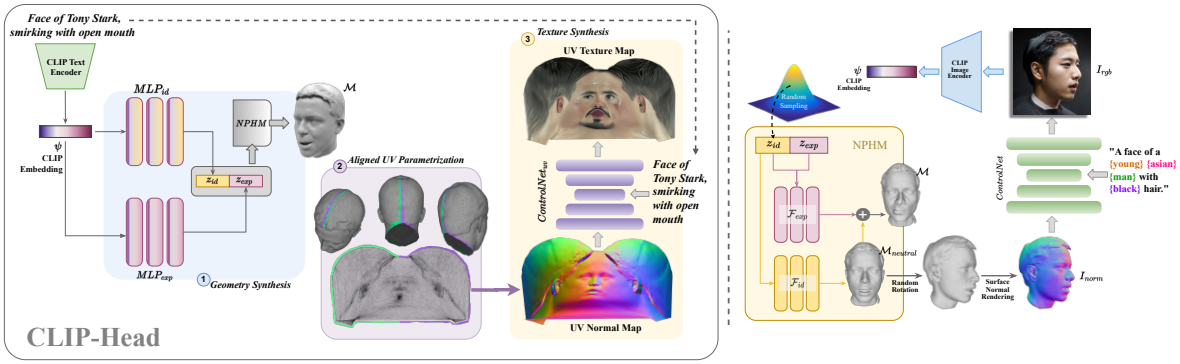


Figure 3.2 Pipeline of the proposed framework *(left)*. Data generation procedure for training MLP_{id} ***(right)*.**

Our work introduced a novel technique for generating 3D textured head meshes from text guidance, capturing varied geometry, expressions, and diverse high-quality textures. Our approach was built upon the NPHM representation, enabling text-driven estimation of its identity and expression latent codes through two mapping networks (MLPs). These MLPs map CLIP text embeddings to the respective NPHM codes and are trained via weak supervision, unlike the supervised approach of HiFi-Face[89], thereby avoiding the need for explicit 3D-text pairs. For appearance generation, we employ advanced CLIP-based models, specifically Latent Diffusion[66], to synthesize high-quality textures guided by text across various head geometries. To ensure consistency between the synthesized texture and the 3D geometry, we propose a custom-trained ControlNet[93]. This architecture leverages UV surface normal maps from the 3D head mesh as spatial control hints for detailed guidance of the UV texture (RGB) map generation. Since 3D scans or generated meshes often possess inconsistent UV layouts, which hinders the creation of effective ControlNet hints, we first introduce a method to approximate UV alignment for 3D head meshes, placing similar semantic regions consistently within the UV space. Finally, during inference, this aligned UV normal map from the generated NPHM mesh guides the ControlNet, producing UV texture maps based on simple text prompts. For an input text prompt, the *Geometry Synthesis* module generates an NPHM head geometry as an SDF, in accordance with the textual description. A mesh can be extracted from the SDF representation using Marching Cubes [74].

Figure 3.3 Clip-Head enables prompt-driven geometry synthesis in a variety of facial expressions and shapes.

The generated mesh is then fed to the *Aligned UV Parameterization module*, which identifies the seam of the head mesh and produces a coherent UV map incorporating projected surface normals. In the final stage, *Texture Synthesis* module generates a UV texture map, guided by the initial text prompt.

$$M = \chi(\mathcal{F}_{id}(z_{id}) + \mathcal{F}_{exp}(z_{id}, z_{exp})) \tag{3.1}$$

3.3.1 Geometry Synthesis

This module's primary objective is to establish a mapping between CLIP's textual embedding space and NPHM's geometric latent space. We introduce an innovative approach using two specialized networks: an identity mapping network (MLP_{id}) and an expression mapping network (MLP_{exp}). These are trained to convert a CLIP embedding vector (ψ) into the respective NPHM latent codes for identity (z_{id}) and expression (z_{exp}). The process begins with a text prompt, which CLIP's encoder transforms into ψ. This embedding is then fed through MLP_{id} and MLP_{exp} to obtain z_{id} and z_{exp}, which are subsequently used in Equation 1 to produce the head mesh M (as shown in Figure 2).

Training the identity mapping network, MLP_{id}, necessitates pairings of ground truth NPHM identity latents (z_{id}) with their corresponding CLIP embeddings (ψ). Lacking a directly annotated dataset for this, we devised a novel automated pipeline to generate these pairs using ControlNet. Illustrated in Figure 3.2 (right), this involves randomly sampling an identity code z_{id} from NPHM's space to create a neutral base mesh with various shapes ($M_{neutral}$). This mesh is then randomly rotated, and its rendered normal map (I_{norm}) is obtained. Guided by a template prompt and this I_{norm} control hint, ControlNet synthesizes an image reflecting the facial features and geometry depicted by I_{norm}. The CLIP embed-

ding (psi) of this synthesized image, along with the initial z_{id}, forms a training pair for MLP_{id}. We generate approximately 10,000 such pairs. This strategy employs weak supervision, implicitly deriving facial shape information through the image decoding process. Consequently, during inference, a text prompt can be converted to ψ by CLIP's encoder, and then mapped by MLP_{id} to an accurate identity latent vector z_{id}, leveraging CLIP's robust text-image alignment.

Training the expression mapping network, MLP_{exp}, uses a distinct strategy to generate (z_{exp}, ψ) pairs. We leverage the dataset from [26], featuring 23 expressions per subject. To maintain separation between our geometry and expression mapping networks, we select expressions from a single identity for this training. Each of the 23 NPHM dataset expressions is first labelled (e.g. happy, sad). For a given expression latent (z_{exp}), we then create varied prompts using the template: **"A face of a [age] [ethnicity] [gender] with [hair color] hair, [expression]."** CLIP's text encoder converts these prompts into embeddings (ψ), which are paired with the corresponding z_{exp} to train MLP_{exp}.

Once both mapping networks are trained, a text prompt's CLIP encoding (ψ) is used by MLP_{exp} to predict z_{exp} and by MLP_{id} to predict z_{id}. These latents are then fed into Equation 3.1 to yield the head mesh M as seen in Figure 3.3

3.3.2 Aligned UV Parameterization

The Aligned UV Parametrization module processes arbitrary head meshes to ensure that semantically similar regions consistently occupy corresponding UV space locations Figure 3.2, which is required for training the Texture Synthesis module, explained in the next section subsection 3.3.3. For a given head mesh M, a low-distortion UV map is achieved by first defining a seam, typically at the mesh's posterior to avoid facial artefacts. A seam can always be considered to run along the backside of the head, as it passes through the axis of symmetry of the human head. Leveraging the shared coordinate system of NPHM meshes obtained via Equation 3.1), a common bisecting plane designates this seam. Subsequently, the mesh's boundary vertices (seam plus neck boundary) are mapped to a canonical 2D curve in the UV space (green/purple curve in Figure 3.2). With this boundary fixed, the UV coordinates for the interior vertices are then derived by computing two harmonic functions (for the U and V coordinates, respectively). Each harmonic function uses the fixed vertices on the curve as boundary constraints to estimate a harmonic parameterization for the remaining vertices.

3.3.3 Texture Synthesis

The final component of our framework addresses the synthesis of a coherent, high-quality UV texture map for the generated head mesh M. Given our goal of creating text-driven UV texture maps (essentially 2D RGB images), leveraging a latent diffusion-based ControlNet appears suitable for producing diverse textures. However, this task presents non-trivial challenges. We propose training a network, $ControlNet_{uv}$, which accepts a spatial control hint and generates a text-guided UV texture map for M. To ensure the synthesised texture accurately follows the mesh's UV layout, this control hint must effec-

tively describe said layout. We achieve this by projecting the face normals onto the UV space, thereby creating a UV normal map to serve as this control hint (as depicted in Figure 3.2). The importance of the previously discussed UV alignment of arbitrary head meshes now becomes clear, as it ensures this control hint possesses semantically meaningful features readily interpretable by $ControlNet_{uv}$. For training $ControlNet_{uv}$, we utilise the same aligned UV parametrisation (from the subsection 3.3.2) on the NPHM dataset's head scans to generate paired aligned UV normal maps and UV texture maps. It is important to note that the original head scans have unstructured, unaligned UV layouts. To automatically create accurate text prompts for this training phase, we render the head scans and use BLIP-2 [43] for image captioning. Once trained, our $ControlNet_{uv}$ can generate high-quality UV texture maps for any given head mesh M, guided by its UV normal map and a user-provided text prompt. While super-resolution can optionally enhance the texture map, all evaluations presented are based on the direct output without this enhancement.

3.3.4 UV Alignment

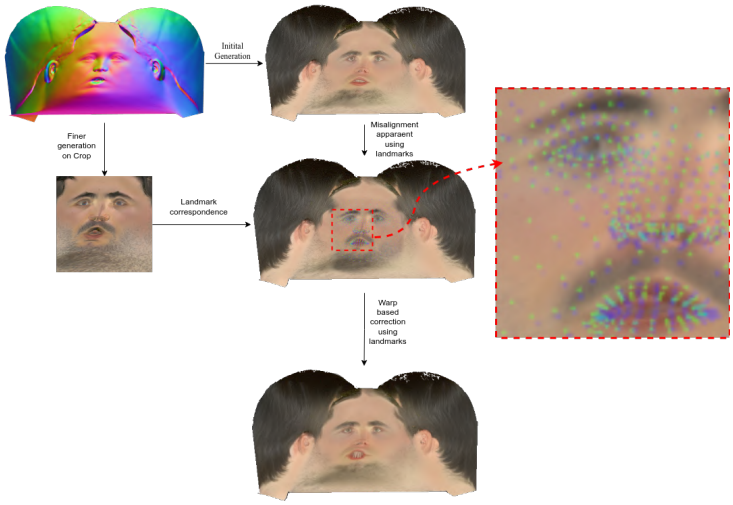


Figure 3.4 Warping module to fix misalignment in the generated texture maps

High-resolution aligned UV texture maps (Figure 3.4) are used for training the Texture Synthesis module. However, due to $ControlNet_{UV}$ 512x512 input limit and a Control strength ¡ 1 (for diversity), the generated texture map edges may misalign with the input UV normal map, especially near facial features (eyes, nose, mouth). We rectify this using a two-stage approach:

1. Generate a full, potentially misaligned UV texture map using the complete UV normal map.

2. Generate a partial, better-aligned UV texture map using a cropped UV normal map of the facial region. Facial landmarks are detected on both maps via MediaPipe [46]. We then warp the full map using TPS [10], mapping its landmarks to the partial map's landmarks (see Figure 3.4).

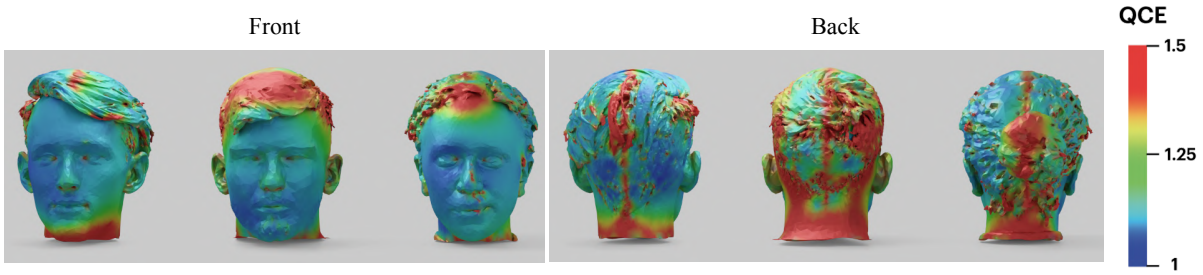


Figure 3.5 Visualization of Quasi-Conformal Error in our common UV Parameterisation

This yields a final UV texture map with improved alignment and retained diversity.

3.4 Evaluation & Results

3.4.1 Training Strategy

Our proposed mapping solution utilized two distinct MLPs to bridge CLIP's latent space with NPHM's latent space. Both MLPs share an architecture consisting of 8 layers, incorporating a single skip connection to the 4th layer. These networks were trained for 100 epochs using the AdamW optimizer [45] on a dataset of 10,000 samples with a batch size of 10. The training objective was to minimize the L2 distance between the MLP-predicted NPHM latent vectors and the corresponding sampled ground truth vectors. All training and subsequent experiments were conducted on an NVIDIA RTX 4090 GPU. Additionally, our implementation of the separate aligned-UV parametrization process leverages the libigl[34] library to compute harmonic weights.

3.4.2 Evaluation Metrics

- **Texture Synthesis Evaluation:** The quality of our texture synthesis is assessed using the CLIP score. This score measures the cosine similarity between the CLIP encodings of the input text prompt and the generated image, utilizing pre-trained CLIP encoders, thereby indicating how well the image reflects the prompt's meaning. Adopting the methodology of [74], we render the textured mesh and compute the CLIP score between this rendered image and the original text prompt. For this evaluation, we use the same text prompts as defined in [74] across ViT-H/14, ViT-L/14, and ViT-B/16 CLIP variants, and our results demonstrate superior performance over state-of-the-art (SOTA) methods.

- **Aligned UV Parametrization Evaluation:** To assess the distortions introduced by our UV parametrization technique, we employ **Quasi Conformal Error (QCE)** [68] and Area Scaling

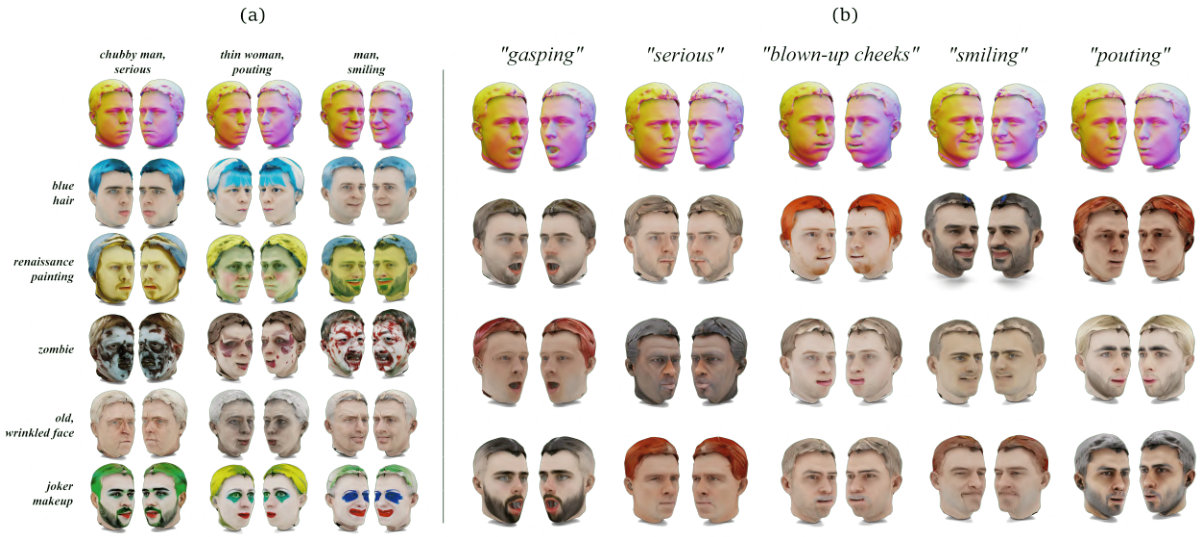


Figure 3.6 Qualitative Results: (a) Text-driven generation and stylization of 3D head meshes. (b) Text-driven generation of 3D head meshes with varying expressions.

Error (ASE) [69]. QCE quantifies angular distortion based on the ratio of singular values from each face mapping, with an ideal value of 1 (higher values indicate more distortion). ASE measures the scaling factor of mapped faces, where negative values mean shrinkage, positive values mean expansion, and zero indicates no area distortion. Our method achieves a median QCE of 4.311, compared to the original UV parametrization's median QCE of 5.143, indicating minimal distortion. The median ASE for our technique is -1.22, while the original default parametrization has an ASE of -0.068. This increase in area scaling is an expected trade-off for achieving the aligned UV parametrization necessary for effective texture synthesis.Figure 3.5 provides a visual representation of QCE on test meshes from the NPHM dataset.

Method	**CLIP Score ↑**		
	ViT-H/14	**ViT-L/14**	**ViT-B/16**
ClipFace	0.287 ± 0.041	0.289 ± 0.039	0.307 ± 0.023
HiFi-Face	0.229 ± 0.033	0.236 ± 0.031	0.300 ± 0.018
CLIP-Head	**0.292 ± 0.035**	**0.303 ± 0.039**	**0.315 ± 0.021**

Table 3.1 Quantitative comparison with SOTA methods.

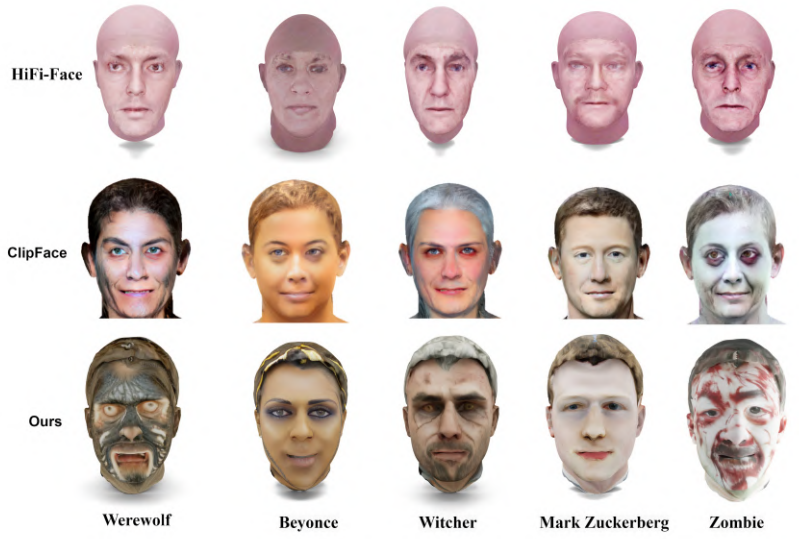


Figure 3.7 Qualitative comparison with existing SOTAs.

3.4.3 Qualitative Evaluation & User Study

Besides quantitative evaluations, we also performed qualitative evaluations, as seen in Figure 3.6, where we show results on a wide variety of prompts. Figure 3.7shows a qualitative comparison with SOTA methods, where our generated results seemed more convincing and true to the input text prompt.

To evaluate the quality of our generated results against existing state-of-the-art methods, ClipFace and HiFi-Face, we conducted a subjective user study involving approximately 70 users. This study comprised two distinct settings. In the first setting, users were randomly shown a generation from one of the three methods and asked to rate how well it matched the input prompt on a scale of 1 to 5. In the second setting, users were presented with outputs from all three methods for the same prompt and asked to select their most preferred result. The average scores for the first setting were **1.38** for HiFi-Face, **2.77** for ClipFace, and **2.85** for our method, CLIP-Head. In the preference study (second setting), CLIP-Head was chosen by 41 users and ClipFace by 29 users, with no users selecting HiFi-Face.

3.5 Discussion

Our proposed method outperformed existing methods in terms of quality and diversity. It can efficiently generate high-quality textured head conditioned from a single text. However, the generated head avatar had a texture, which might have light baked into it. Therefore, the generated head avatar could not be relit under new lighting. A natural next step to improve this procedure is to generate a texture map, where the appearance is disentangled from lighting conditions, such that it could be relit into another environment. However, lack of ground truth data makes it a difficult problem.

Our method generates a prompt-adhering and high-quality texture map, but the generated texture map is static. For instance, the texture map may appear realistic for the given expression, but when the same

texture map is used for a different expression, even for the same face shape, it may not appear realistic, as some facial features like wrinkles are expression-dependent. A dynamic expression-dependent texture map is essential to accurately represent these features.

Our choice of parametric head models can represent hair to some extent. However, hair strands are not modelled individually, making them a part of facial geometry. This kind of hair geometry cannot be physically simulated, which is a requirement in game engines. Hair generation is also a problem statement which has not been fully explored yet.

3.6 Summary

Our work proposed a novel method for generating textured 3D head models directly from text descriptions, aiming to overcome limitations in geometric control and texture diversity/consistency found in prior approaches. We uniquely combined the expressive, disentangled NPHM representation for geometry with modern generative techniques. Our key innovations included two weakly-supervised mapping networks (MLPs) that translate CLIP text embeddings into NPHM's identity and expression latent codes, enabling intuitive text control without requiring paired 3D-text data. For high-quality appearance, we employed Latent Diffusion steered by a custom-trained ControlNet. Crucially, we introduced a novel UV alignment technique for NPHM meshes, allowing us to create consistent UV normal maps that serve as precise conditional hints for our ControlNet, ensuring the synthesised textures align accurately with the 3d geometry. Our integrated framework achieved single feed-forward pass generation of diverse 3D heads with detailed geometry, expressions, and corresponding high-fidelity textures, driven solely by text prompts.

To overcome the constraints of baked-in lighting on textured head avatar appearance and the significant lack of accessible lightstage-based datasets, the following chapters propose two key contributions. We will detail a novel Gaussian head-based representation engineered for relightability and introduce a novel, scalable capture setup leveraging smartphones, which is intended to help bridge the critical data scarcity.

Chapter 4

LightHeaded: Relightable & Editable Head Avatars from a Smartphone

In the previous chapter, we proposed an approach that produced a head avatar that had lighting information baked into it; that is, the generated head cannot be accurately relit when placed in a new environment. To generate a head with a disentangled appearance, we require a dataset that has albedo texture maps, normal maps and specular/roughness maps, which could be used to accurately relight a head using a shading model. However, no such dataset is publicly available. Therefore, in the subsequent chapters, we will focus on methods to create realistic head avatars.

The conventional method for creating photorealistic, animatable, and relightable 3D head avatars, using expensive Lightstage setups with multiple calibrated cameras, limits their widespread accessibility. To address this, we introduced a novel, cost-effective solution for generating high-quality relightable head avatars using only a smartphone fitted with polaroid filters. Our technique simultaneously captures cross-polarized and parallel-polarized video streams of dynamic facial performances in a controlled dark room environment with a single point-light source, enabling the separation of the skin's diffuse and specular reflectance components. We propose a hybrid avatar representation that embeds 2D Gaussians within the UV space of a parametric head model, achieving both efficient real-time rendering and high-fidelity geometric detail. Furthermore, our learning-based neural analysis-by-synthesis pipeline disentangles pose and expression-driven geometric offsets from appearance, decomposing the surface

Figure 4.1 LightHeadEd: A Textured Gaussian Head Avatar with animation, relighting & editing support.

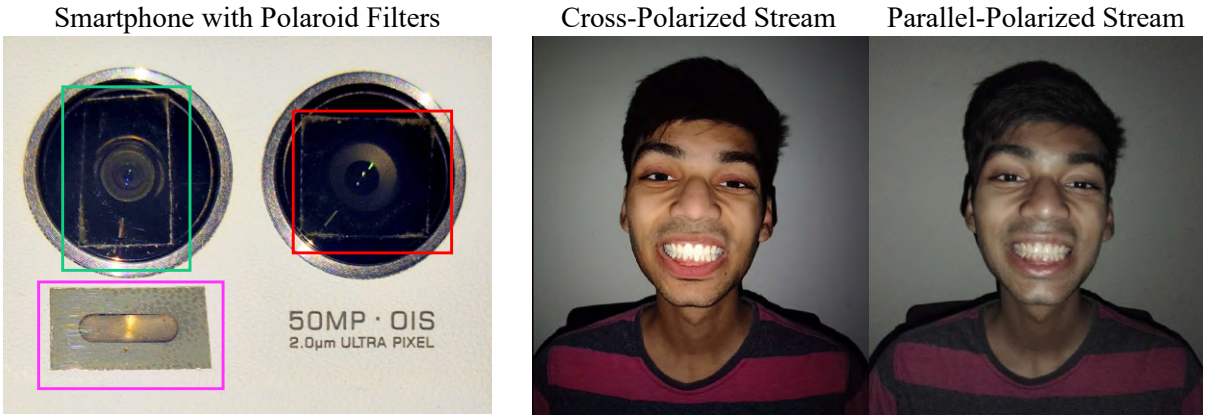


Figure 4.2 Proposed dynamic capture setup: Smartphone equipped with polaroid filters (left); Cross-Polarized & Parallel-Polarized Monocular Video Streams (right).

into albedo, normal, and specular UV texture maps, as well as environment maps. As part of this work, we also gathered a unique dataset featuring various subjects exhibiting diverse facial expressions and head movements.

Our novel capture method and representation could be further used to create a dataset of relightable head avatars, which could be used to train a generalizable relightable head model.

4.1 Introduction

Relightable 3D human head avatars are increasingly popular as immersive communication tools in platforms like the Metaverse, mixed reality, and telepresence. However, producing personalized avatars with these relighting capabilities is difficult because it typically requires extensive multi-view data from sophisticated, expensive, and computationally demanding volumetric light-stage systems, usually only accessible to high-budget studios. In contrast, developing a method to create affordable, realistic, animatable, and relightable head avatars using just a standard smartphone would democratize access for end-users and open up numerous applications in communication and entertainment.

While some existing methods [12, 42] aim to generate personalized 3D head avatars from monocular RGB videos (e.g., from smartphones/webcams), they predominantly rely on universal prior models. These priors are trained on comprehensive light-stage datasets of many individuals to separate facial geometry and reflectance. This strong reliance on large datasets and pre-trained models compromises affordability, limits generalizability to new demographics and appearances, and often necessitates costly preprocessing and model fine-tuning at test time. To avoid dependency on such priors, alternative approaches [6, 95, 24] utilize inverse physically-based rendering (PBR) for self-supervised, analysis-by-synthesis reconstruction of personalized head avatars. For instance, method [6] utilizes the FLAME

parametric head model [44] to reconstruct a relightable mesh from several unconstrained RGB videos of an individual. While FLAME allows animation via low-dimensional pose/expression parameters, it only offers a coarse geometric approximation and fails to capture surface deviations like skin deformations or hair. To address this, recent approaches [47, 14] combined 3D Gaussian Splatting (3DGS) [39] with a FLAME mesh, sampling 3D Gaussian splats over it to learn person-specific details while retaining pose/expression control. However, these methods tend to be resource-heavy, lack relighting capabilities, and can produce inaccurate geometry. Furthermore, existing monocular methods generally do not offer control over appearance manipulation.

In this work, we introduced a novel approach to easily create animatable, editable, and relightable head avatars using only a commodity smartphone. Moving away from reliance on calibrated lightstages for dynamic facial performance capture, we propose an affordable, calibration-free process to collect realistic head deformation and facial reflectance data. To facilitate relighting, animation, and editing, we present a new **textured Gaussian head** avatar representation coupled with an effective self-supervised training methodology. This allows us to learn a personalized avatar with high-quality geometry and decomposed albedo, normal, and roughness UV maps. Our inspiration comes from lightstage setups, which use multiple RGB cameras with polaroid filters to separate the skin's diffuse and specular responses. Emulating this, we equipped a smartphone's dual cameras and flashlight with an inexpensive polaroid film Figure 4.2, creating a scalable method to capture realistic head deformations with decomposed surface illumination from monocular videos.

We used this captured polarization data to reconstruct a temporally consistent, relightable 3D head avatar. After tracking a FLAME mesh across the monocular videos, we employ 2DGS [33] to model details like skin and hair. Unlike methods [47, 14] that combine 3DGS with FLAME, our 2DGS representation allows direct surface regularization via normal constraints, leading to high-fidelity facial details and real-time rendering. We embed these 2D Gaussian disks into FLAME's UV space, learning Gaussian attributes as UV maps for straightforward animation and texturing. Storing appearance in UV space obviates memory-intensive SH coefficients and, crucially, enables flexible appearance editing by modifying the albedo map. We decouple reflectance into albedo, normal, and roughness UV maps, and also learn expression-dependent residual UV maps to handle dynamic changes, alongside an environment cubemap for relighting in varied conditions.

Our proposed method for avatar capture and reconstruction is highly efficient, offering affordable relightable heads with compelling features. We demonstrate superior results through comprehensive quantitative and ablative analyses and showcase useful applications like shape editing and text-guided appearance editing.

4.2 Related Work

Lightstage Capture Systems: Several works have long utilized polarization to decompose scene illumination [53, 62, 87], capitalizing on how single-bounce specular reflection preserves incoming

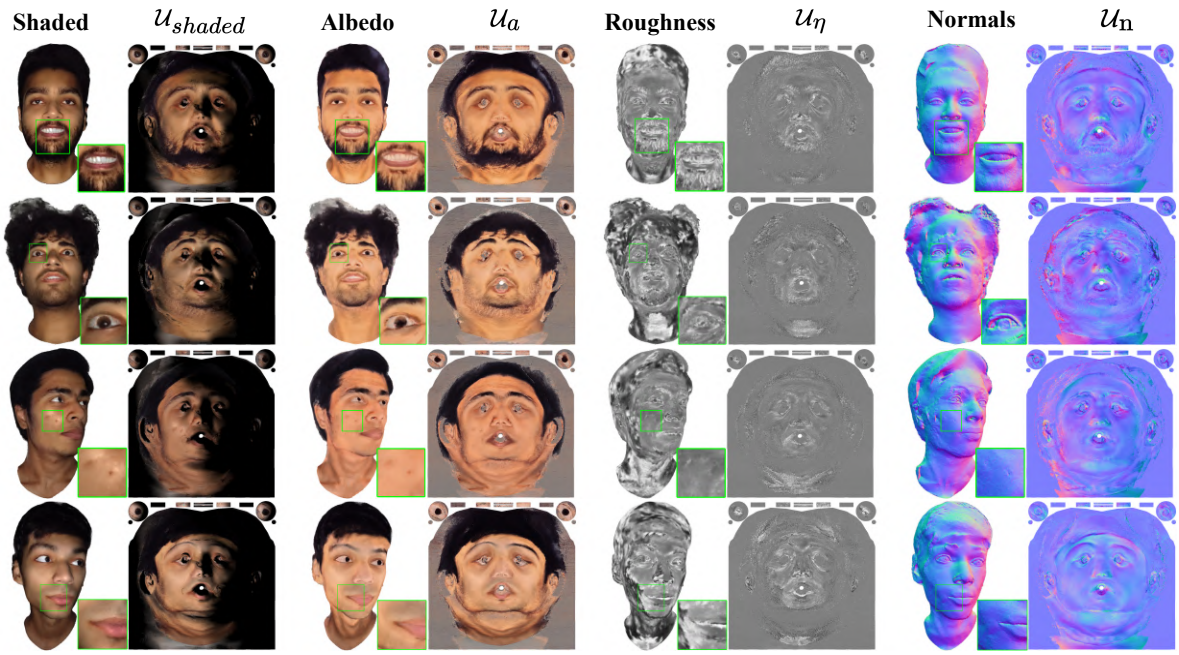


Figure 4.3 Decomposition of appearance & geometry in UV space.

light's polarization. Debevec et al. [18] pioneered the first Lightstage by using polarization filters and controlled lighting to estimate human face reflectance, effectively capturing appearances under all lighting directions. While initially designed for static subjects, later advancements [25, 86, 88] introduced motion compensation techniques, broadening capture areas and enhancing surface fidelity. Method [28] suggested a multi-view system for dynamic facial texture capture without polarized light. Subsequently, [64] brought back polarization, this time without active illumination, to model subsurface scattering. More recently, [92] further refined these systems to incorporate global illumination and polarization modelling. Although these volumetric lightstage solutions yield remarkable visual results for face reflectance and geometry, their dependence on sophisticated, expensive, and bulky hardware, coupled with high operational expertise, presents scaling challenges for diverse identities.

Data-driven Personalized Head Avatars: Control over facial expressions has evolved from blendshape techniques in VFX/gaming to data-driven methods estimating statistical bases from 3D scans, allowing detailed head pose and expression manipulation [8, 57, 11, 82, 63, 81]. Yet, models such as FLAME [44] often overlook subtle expressions and unique person-specific geometry (like hair or skin details). For photorealism, many methods utilize differential rendering from multi-view videos within an analysis-by-synthesis pipeline [24, 41, 94, 40, 60, 27, 91]. NeRF-based personalized avatars, while achievable, face slow rendering and coarse geometry issues. This has led to recent use of 3D Gaussian splats (3DGS) [39] for real-time efficiency, though often with baked-in illumination that precludes relighting. Some works [67, 42] produce relightable avatars using lightstage data, but this limits demographic diversity. While

diffusion-based [48, 79, 13] and neural editing techniques [30, 50] allow texture modification, they are often slow and challenged by new poses. A recent method [84] employs neural texture maps for editing flexibility but involves optimization-heavy steps. Our paper introduces an affordable, scalable approach for creating animatable, relightable, and editable head avatars.

Head Avatars from Monocular Video(s): Generating animatable head avatars from monocular videos presents another intriguing yet challenging research avenue. Many current learning-based techniques try to bypass the need for multi-view data by tracking head topology changes over a single video, representing it either as a parametric mesh [6] or a neural head model [26]. To achieve more detailed and expressive results from monocular input, some methods use primitives like points [95] or 3D Gaussians [14] layered onto a FLAME [44] base. Approaches such as [72] and Gaussian Blendshapes [47] utilize Gaussian splats as primitives, but struggle with high-quality surface details due to the ill-defined normals inherent to 3D Gaussians. These methods also typically lack any support for editing the head meshes. PointAvatar [95] attempts to reconstruct relightable avatars under very restricted, fixed lighting conditions; while visually impressive, its relighting isn't physically based and ignores skin reflectance properties. Additionally, point primitives are memory-heavy, rendering [95] unsuitable for real-time use. A significant general drawback is that existing monocular head avatar methods do not directly output texture maps, which are crucial for flexible and quick appearance editing. Although FlashAvatar [90] does embed 3D Gaussians in FLAME's UV space for initialization, it ultimately doesn't learn UV texture maps and lacks relighting capabilities.

4.3 Preliminary Proposed Framework

To enable the creation of affordable, person-specific, relightable head avatars from monocular input, we introduced an effortless polarized data acquisition process. This captured data then formed the basis for a novel textured head avatar representation. For controlling head pose and expressions, we utilized FLAME, augmenting it with 2DGS [33] to model individual-specific offset details and appearance. To achieve relighting capabilities, we propose an innovative self-supervised learning scheme. This scheme decomposes the captured polarized information into distinct appearance and shading components, represented as albedo, normal, and roughness UV maps within FLAME's parametrization space. Furthermore, to account for dynamic appearance changes linked to pose and expressions, we learned an expression-conditioned residual albedo map. The polarized data capture method, the proposed head avatar representation, and the self-supervised training methodology are discussed ahead.

4.3.1 Polarized Capture Setup

In contrast to method [5], which uses a single back camera to sequentially capture cross and parallel-polarized visuals of a static person (requiring them to remain still for extended periods), we introduce a more casual, dynamic, and scalable capture process. Our setup involves a single smartphone mounted

on a tripod in a dark room, with its flashlight serving as the sole point-light source to simplify scene lighting estimation. As illustrated in Figure 4.2, a thin, linearly polarized filter (approximately 4.8 microns thick, highlighted in purple) covers the flashlight. The smartphone's two back cameras are also covered with the same type of polarised film, but with a relative angular difference of 0 (red) and 90 (green) with respect to the flashlight's filter. This configuration ensures one camera is parallel-aligned and the other is cross-aligned to the flashlight's polarization.

Light is an Electromagnetic (EM) Wave, where the plane of oscillation of the electric field vector is perpendicular to the magnetic field oscillation. Polarized light is one in which the oscillations of the electric field vectors are restricted to a single plane [31].

A linear polarized filter allows light to pass through it if it is of a specific polarization, while blocking other polarizations. In the context of our setup, we have a linear polariser on the flashlight. Therefore, the light coming out of it is polarized along a certain plane. One of the camera lenses has a polarizer, with the same orientation as the flashlight, which implies that if light is polarized in the same direction as the one coming out of the flashlight, it will be captured. The other lens has a polarizer in an orientation that is perpendicular to the orientation of the polarizer on the flashlight. This means that no light from the flashlight will pass into it, unless the polarisation is changed. We exploit this fact to capture how various layers interact with the skin.

When light is reflected off the oily layer of the skin, the polarization of light is preserved, and the specular details of the face are visible. Since polarization is preserved, these specular details can be captured by the lens which has a polarizer alignment similar to the polarizer on the flashlight, referred to as parallely polarized. On the other hand, when light is scattered within the skin, it loses its polarisation. Therefore, both cameras can capture components of unpolarized light that align with their respective polarisers. This reflected unpolarized light is mainly visible in the lens, which has a polariser alignment perpendicular to that of the flashlight, referred to as cross-polarized. Thus, the cross-polarized lens is able to capture diffuse reflections that occur at the skin. As a result, in the two simultaneous streams, we are able to concurrently capture the diffuse and specular reflections off of the skin.

The use of polaroid filters results in a noticeable tint shift between the two camera streams (see Figure 4.2). While [5] addresses this by precomputing an approximate affine color correction matrix, we defer tint-shift correction to the training phase (discussed in subsection 4.3.5) for a more seamless capture experience.

Using this setup, we simultaneously record two continuous monocular videos (one cross-polarized, one parallel-polarized) of a human subject performing a predefined set of diverse expressions and poses. This ensures we capture adequate deformation and lighting information from various angles. Both video streams are recorded at 1920x1080 resolution and 24 FPS. Crucially, although we use two cameras simultaneously for synchronization, we do not employ any stereo-based depth or alignment information; instead, we treat the two streams as separate monocular RGB videos to facilitate avatar creation. Our capture process is further demonstrated in the supplementary video.

4.3.2 Textured Gaussian Head Representation

From the captured cross-polarized and parallel-polarized video sequences, we first obtain a subject-specific FLAME head mesh (M) that is temporally tracked over both monocular sequences, using the head-tracking pipeline [59] from [61]. To model the remaining subject-specific details beyond this base FLAME mesh, we utilize 2D Gaussian Splats (2DGS) [33], essentially flat, 2D planar elliptical disks embedded in 3D space. 2DGS employs an explicit ray-splat intersection method, which results in perspective-correct splitting and thereby improves surface reconstruction quality. This approach also facilitates direct surface regularization via normal constraints [33], enhancing the quality of normals and subsequent shading. Each 2D Gaussian disk (G_i) is defined by its 3D mean position ($p_i \in \mathbb{R}^3$), 2D scale ($S_i \in \mathbb{R}^2$), orientation quaternion ($q_i \in \mathbb{R}^4$), and opacity ($o_i \in \mathbb{R}^1$). Similar to 3DGS [33], the factorized covariance matrix (Σ_i) for each 2D Gaussian G_i is given by

$$\Sigma_i = R_i S_i' S_i'^{\top} R_i^{\top} \tag{4.1}$$

where R_i is the rotation matrix from q_i, and $S_i' = [S_i, 0]^{\top} \in \mathbb{R}^3$ (the third scale dimension is zero for a flat 2D Gaussian).

Our key novelty, however, is associating all learnable attributes of these 2D Gaussians with FLAME's UV space. Instead of embedding them in 3D object space, we embed them in the tangent space of the triangulated mesh. This means that rather than initializing each 2D Gaussian's mean position p_i as (x_i, y_i, z_i) in 3D, we initialize it with (u_i, v_i) coordinates within FLAME's canonical UV space $\mathcal{U}$ (as shown in Figure 4.4). We also precompute a face index map ($\mathcal{U}_{idx}$) that stores the triangle index (m) corresponding to each UV coordinate in $\mathcal{U}$ (this is a one-time step). To transform these UV-space Gaussians from the canonical pose/expression to the deformed pose/expression of M, we first find the associated face index $m = \mathcal{U}_{idx}(u_i, v_i)$. This gives us the 3D vertex positions of the triangle $t_m = (v_{m0}, v_{m1}, v_{m2})$. The 3D mean position p_i along the surface of the mesh is then simply computed as $p_i = (\alpha_i * v_{m0}) + (\beta_i * v_{m1}) + (\gamma_i * v_{m2})$, where $\alpha_i, \beta_i, \gamma_i$ are the barycentric coordinates for (u_i, v_i) (and $\alpha_i + \beta_i + \gamma_i = 1$). This allows direct estimation of each Gaussian G_i's location in the posed 3D space. This novel formulation enables us to define all Gaussian attributes (orientation, scale, offsets, appearance, etc.) as learnable UV maps.

4.3.3 Gaussian Attributes a UV Maps:

Building on the aforementioned formulation, we now detail how Gaussian attributes are modelled as texture maps. For every valid UV texel (excluding empty regions), we sample 2D Gaussians and define distinct UV maps for various attributes—$\mathcal{U}_a$ (albedo map), $\mathcal{U}_\eta$ (roughness map), $\mathcal{U}_T$ (tangent map), $\mathcal{U}_B$ (bitangent map), $\mathcal{U}_\delta$ (offset map), $\mathcal{U}_s$ (scale map), and $\mathcal{U}_o$ (opacity map). As illustrated in Figure 4.4, for any Gaussian G_i at UV coordinate (u_i, v_i), we can retrieve its attribute θ_i using $\mathcal{U}_\theta(u_i, v_i)$, where θ_i represents attributes like $a_i, \eta_i, T_i, B_i, s_i$, or o_i.

To determine the orientation of each 2D Gaussian G_i, instead of directly using a quaternion, we query its tangent $T_i = \mathcal{U}_T(u_i, v_i)$ and bitangent $B_i = \mathcal{U}_B(u_i, v_i)$. The normal N_i is then calculated

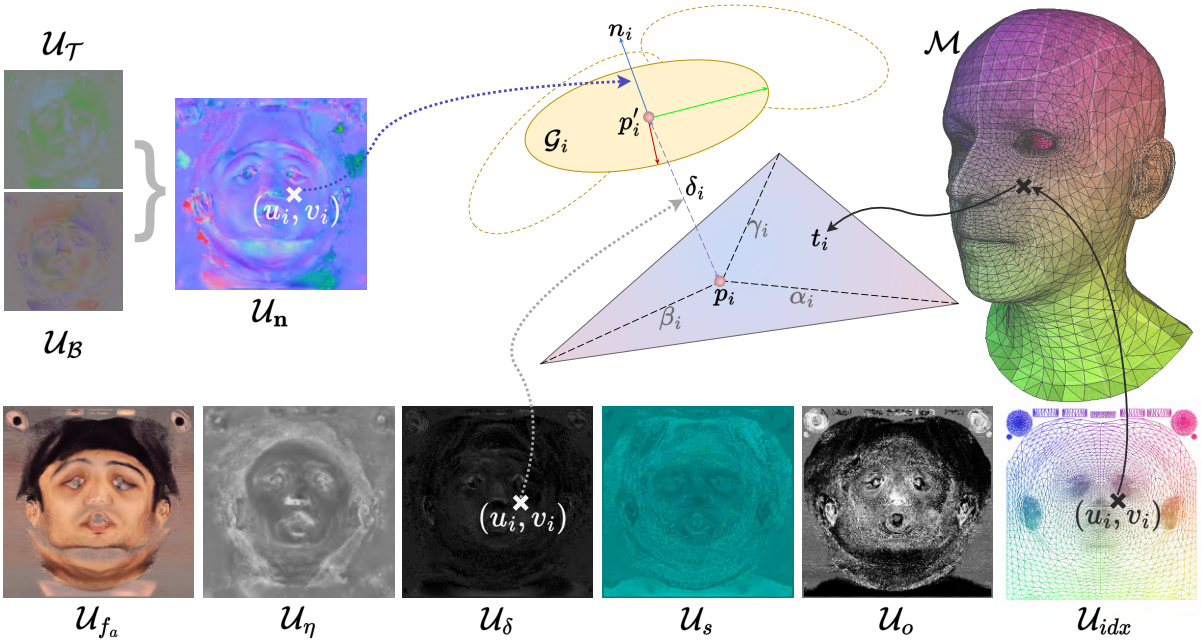


Figure 4.4 Textured Gaussian head representation.

as their cross product: $N_i = T_i \times B_i$. We then employ a precomputed TBN matrix to transform this normal n_i (derived from N_i) to estimate a smooth rotation R_i in the 3D object space as:

$$R_i = \text{TBN} * n_i \tag{4.2}$$

To incorporate geometric details that deviate from the surface of the parametric head mesh M, we query an offset $\delta_i = \mathcal{U}_\delta(u_i, v_i)$ along the normal direction. This offset is used to shift the initial mean position p_i of the 2D Gaussian in 3D space, resulting in a new position p_i':

$$p_i' = p_i + \xi_i * R_i \delta_i \tag{4.3}$$

where ξ_i is an area-adjustment factor calculated as:

$$\xi_i = (\alpha_i * a_{m0}) + (\beta_i * a_{m1}) + (\gamma_i * a_{m2}) \tag{4.4}$$

and

$$a_{mj} = \frac{1}{|N(v_{mj})|} \sum_{k \in N(v_{mj})} \sqrt{\text{Area}(t_k)} \tag{4.5}$$

Here, α, β, γ are barycentric coordinates, and a_{ij} for vertex v_{ij} is the mean root-sum of the areas of triangles in the neighborhood $N(v_{mj})$ of v_{mj}. The factor ξ_i prevents Gaussians associated with smaller triangles from moving too far or scaling excessively. Finally, the resulting 2D Gaussian position p_i', its shaded color c_i, and opacity o_i are utilized in the splatting-based rendering process.

4.3.4 Rendering Equation & BRDF:

For modelling appearance, the original 2DGS method employs Spherical Harmonics (SH)[23] to learn the visual characteristics of each 2D Gaussian. While SH coefficients can accurately represent view-dependent appearance in static scenes, modelling high-frequency details (such as those from surface normals and roughness) necessitates higher-order SH coefficients, which exponentially increase memory demands. Crucially, direct appearance editing is not feasible with such SH-based representations.

These limitations motivate our decision to replace SH coefficients with a single RGB color, c_i, for each Gaussian's appearance. To achieve view-dependent color, we implement a novel physically-based SVBRDF [55] shader specifically for 2DGS. This shader estimates the Gaussian's appearance based on the viewing direction 'ω' and the light direction 'l' from a point source (e.g., a flashlight). Furthermore, to enable relighting by learning appropriate shading, we disentangle the appearance c_i into its albedo (a_i), specular (f_{si}), and diffuse (f_{di}) components. For the specular component f_{si}, we utilize the Cook-Torrance[16] microfacet specular BRDF, defined as:

$$f_{si}(l, \omega, \eta_i) = k_s * \frac{D(h)F(\omega, h)G(l, \omega, h)}{4(n \cdot l)(n \cdot \omega)} \tag{4.6}$$

where h is the half-vector bisecting the angle between l and ω, and k_s is a constant specular gain. Consistent with [5], we employ Schlick's approximation [70] for the Fresnel term F. For the Normal Distribution Function (NDF) term D, we adopt an alternative approximation from Trowbridge-Reitz[83], while our geometric term G uses Smith's variant of the Schlick-GGX approximation [70]. Moreover, unlike [5], which uses a spatially varying specular gain k_s (a learnable scalar in our implementation), we instead use a spatially varying roughness η_i associated with each Gaussian G_i.

$$D(h, \eta_i) = \frac{\eta_i^2}{\pi((n \cdot h)^2(\eta_i^2 - 1) + 1)^2} \tag{4.7}$$

$$G(l, \omega, h) = G^2_{\text{Schlick}}(\omega) = \left(\frac{n \cdot \omega}{(n \cdot \omega)(1 - \lambda) + \lambda}\right)^2 \tag{4.8}$$

where $\lambda = \eta_i/2$ for remapping Schlick-GGX to align with Smith's formulation [76]. It is important to note that, similar to [5], we assume $l = \omega$ because the flashlight is positioned very close to the camera lens, which simplifies the implementation. For the diffuse component, we use the BRDF model proposed by Ashikhmin & Shirley[3]:

$$f_{di}(a_i, \omega) = \frac{28a_i}{23\pi}(1 - F_0)\left(1 - \left(1 - \frac{n^\top \omega}{2}\right)^5\right)^2, \tag{4.9}$$

where a_i is the albedo color, and $F_0 = 0.04$ represents the skin's reflectance at normal incidence. The final shaded colour is then computed as:

$$c_i = f_{di} + f_{si} \tag{4.10}$$

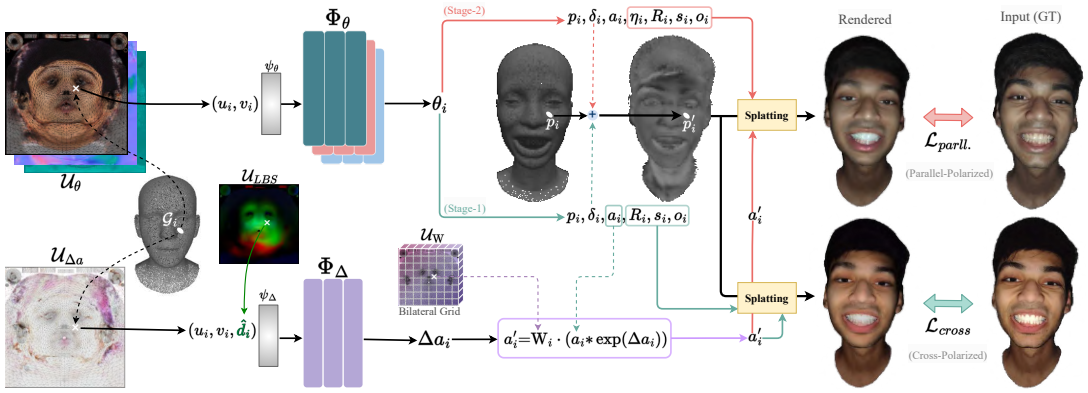


Figure 4.5 Proposed two-stage training strategy to learn textured Gaussian head avatars with decomposed appearance and geometry.

4.3.5 Learning 2D Gaussian Attributes

Figure 4.5 illustrates our novel training methodology for learning a textured Gaussian head avatar. Given input cross-polarized and parallel-polarized video sequences, we learn the aforementioned Gaussian attributes through a two-stage process. We utilize a set of stacked Multi-Layer Perceptrons (MLPs), denoted as Φ_θ, which comprises several hash-encoded MLPs [52] (one dedicated to each attribute) utilizing hash encoding ψ_θ. For learning pose-specific albedo, we employ an additional hash-encoded MLP, Φ_Δ, with its corresponding hash encoding ψ_Δ. To address the global tint-shift observed between the cross-polarized and parallel-polarized video frames, we incorporate a low-dimensional Bilateral Grid, $\mathcal{U}_W$, which has a learnable affine transformation matrix W_i for each UV texel. Furthermore, because we allow the (u_i, v_i) coordinates to be optimized over time, we also precompute a triangle-index map, $\mathcal{U}_{idx}$, for efficient lookup of triangle indices from UV coordinates. All attributes specified within $\mathcal{U}_\theta$, $\mathcal{U}_\Delta$, and $\mathcal{U}_W$ are initialized randomly. They are then optimised via differential rendering losses in a self-supervised manner, using the cross-polarized video sequence during the first stage and the parallel-polarized sequence during the second stage.

We start by initializing 2D Gaussians, G_i, for each UV texel with coordinates (u_i, v_i), enabling us to learn their attributes directly embedded within the UV space. In the first stage of training, our primary focus is on learning the attribute maps $\mathcal{U}_\theta$ (excluding the roughness map $\mathcal{U}_\eta$) and the pose-specific albedo residual map $\mathcal{U}_{\Delta a}$. The UV coordinates (u_i, v_i) are fed as input to the hash-encoded MLP Φ_θ to predict the attribute set $\theta = \Phi_\theta(\psi_\theta(u_i, v_i))$. From this, we extract or estimate the albedo a_i, orientation R_i, scale s_i, opacity o_i, and the shifted mean position p'_i (as per Equation 4.3).

For pose and expression conditioning, we utilize the Linear Blend Skinning (LBS)-based displacement of the FLAME mesh, denoted as $\hat{d}_i = \mathcal{U}_{\text{LBS}}(u_i, v_i)$. This displacement is then used to predict a pose-specific residual albedo $\Delta a_i = \Phi_\Delta(\psi_\Delta(u_i, v_i, \hat{d}_i))$. The tint-corrected albedo is subsequently

computed as:

$$a_i' = W_i \cdot (a_i * \exp(\Delta a_i)) \tag{4.11}$$

where the transformation matrix W_i is set to the identity matrix during this stage, as we are processing cross-polarized data. Using this corrected albedo a_i', we then calculate the final shaded color c_i using Equation 4.10 (while ignoring the specular component at this point). This color c_i, along with the orientation R_i, scale s_i, and opacity o_i, is used to perform 2D Gaussian splatting to render the image, and we minimize the loss L_{cross}.

In the second training stage, we cease optimization of the base and residual albedo maps ($\mathcal{U}_a$ and $\mathcal{U}_{\Delta a}$) and concentrate on learning the roughness map, while simultaneously fine-tuning the remaining attributes using supervision from the parallel-polarized data. Similar to the first stage, we predict the attribute set θ, from which we extract or estimate the albedo a_i, orientation R_i, scale s_i, opacity o_i, and the shifted mean position p_i'. A key difference in this second stage is that we also learn the bilateral grid $\mathcal{U}_W$. This is necessary because a tint shift exists between the albedo learned in the previous stage (from cross-polarized data) and the parallel-polarized images. We obtain the tint-corrected albedo a_i' by plugging the values of W_i (from $\mathcal{U}_W$) and a_i into Equation 4.11.

During this second stage, both diffuse and specular components are included when computing the shaded color c_i (as per Equation 4.10). Using these estimated Gaussian attributes, we render an image via splatting and minimize the loss L_{parll}. Both L_{cross} (from the first stage) and L_{parll} are defined as:

$$L_{\text{cross/parall.}} = L_1 + L_{\text{SSIM}} + L_{\text{LPIPS}} + L_{\text{scale}} + L_{\text{flame}} \tag{4.12}$$

where $L_{\text{scale}} = \sum_i \max(0, s_i - 0.3)$ serves to prevent the Gaussians from becoming excessively large, and L_{flame} is a FLAME-based regularization term. For L_{flame}, we rasterize the FLAME mesh and compute L_1 and L_{SSIM} losses between this rendered FLAME image and the input image.

For inference, we discard the MLP Φ_θ and instead use the attributes directly stored in $\mathcal{U}_\theta$ for efficient rendering. However, we continue to rely on Φ_Δ to model appearance changes corresponding to novel poses or expressions.

4.4 DuoPolo Dataset

We also presented the "DuoPolo" dataset, a collection comprising cross-polarized and parallel-polarized video sequences from approximately 10 subjects, all captured using our proposed setup. Each data sample within this dataset includes a 90-second video sequence where a human subject performs a variety of head poses and facial expressions, concluding with a short spoken phrase to capture subtle lip movements during speech. Every sequence is recorded at 1920×1080 resolution and 24 FPS. In addition to the video data, we provide per-frame background/foreground segmentation masks, 3D facial landmarks, and the parametric FLAME[44] head mesh, which has been tracked over both video streams using the method from[59]. This proposed dataset is the first of its kind, aiming to make polarized facial

performance data widely accessible, thereby bridging the gap between costly light-stage acquisitions and the creation of affordable head avatars.

4.5 Experiments & Results

For our experimental evaluations, we utilized sequences sourced from the "DuoPolo" dataset that we introduced. Each sequence was divided into training and testing sets, with the initial 80% of frames allocated for training and the subsequent 20% reserved for testing. To quantitatively assess performance, we adopted standard metrics prevalent in existing state-of-the-art methodologies. Specifically, following the evaluation protocol outlined in[95], we calculated the per-sequence (or per-subject) L_1 error, Peak Signal-to-Noise Ratio (PSNR)[19], and Structural Similarity Index Measure (SSIM)[32]. These metrics are computed by comparing the rendered frames generated by our method against the corresponding input (ground truth) frames from the designated test split.

4.5.1 Qualitative Results

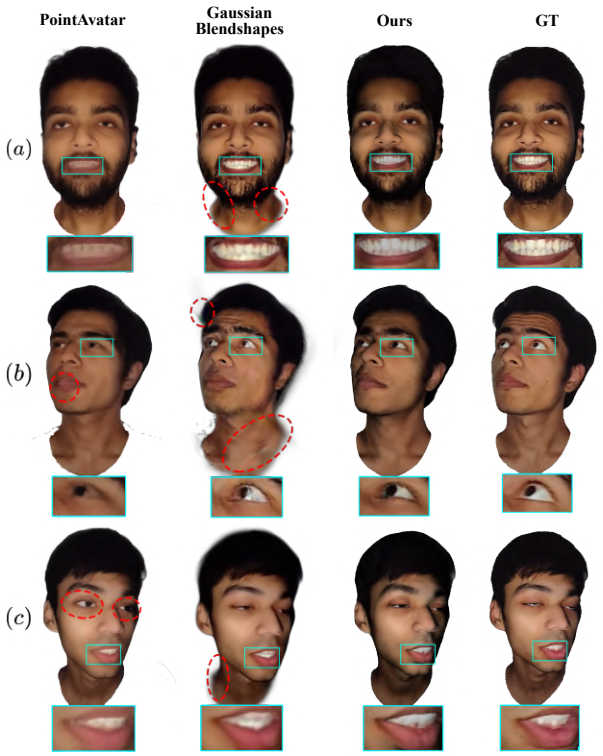


Figure 4.6 Qualitative comparison with SOTA methods. Our method is able to better capture finer details (such as teeth and eyes) whilst supporting relighting, pose and expression control.

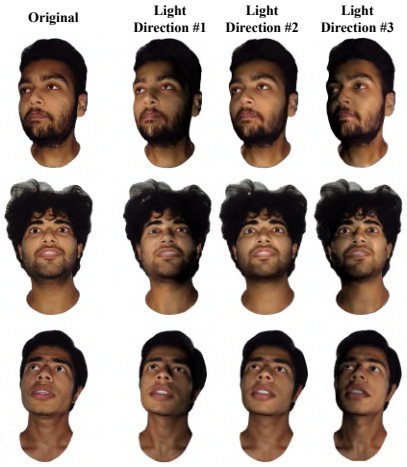


Figure 4.7 Relighted novel head poses & expressions results.

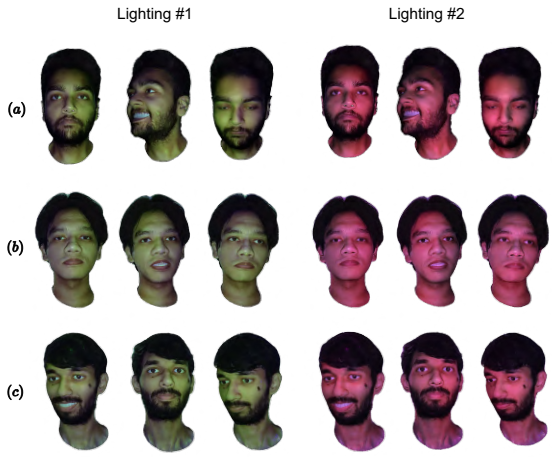


Figure 4.8 Additional Rendering the head avatars under different lighting.

Surface Reflectance Map: Qualitative results demonstrating our learning methodology are presented in Figure 4.3, showcasing outputs from a few sequences acquired with our proposed capture setup. As evident, our technique can effectively model physically accurate facial reflectance by successfully disentangling appearance information (in the form of albedo, roughness, and normal UV maps) from the geometry. This disentanglement leads to high-quality renderings that also preserve high-frequency geometric details.

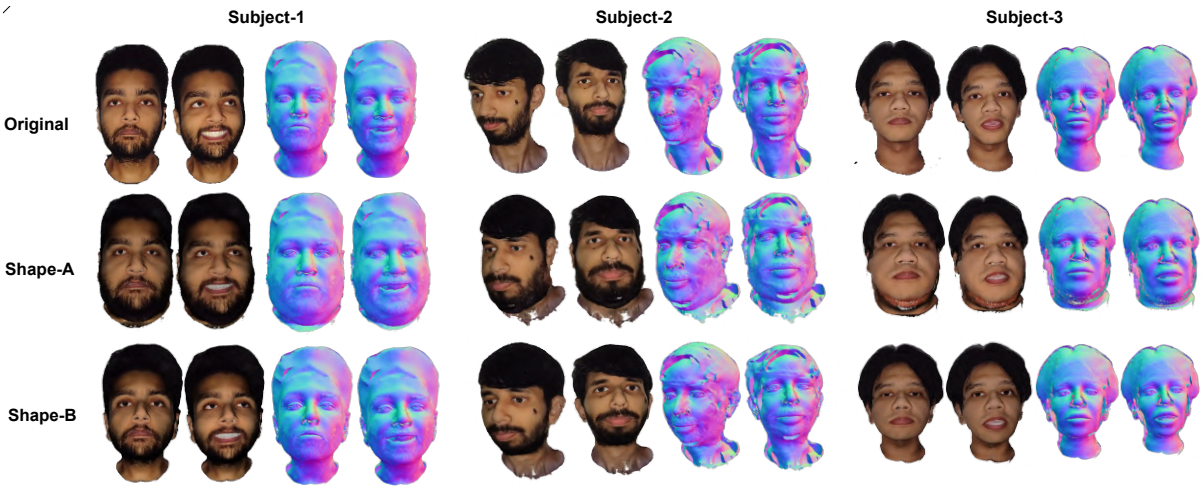


Figure 4.9 Shape Editing

Relighting: Figure 4.7 and illustrates the relighting capabilities of our learned head avatars. Here, we demonstrate relighting the avatars from various light directions while they are in a novel pose, which is

referenced from an original test frame. This highlights our representation's ability to achieve view and pose-consistent relighting.

Appearance Editing: The textured human head representation we proposed facilitates appearance editing through direct modification of the albedo texture map, akin to traditional textured mesh editing techniques. This allows users to change the avatar's appearance and then re-render it. An example of this application is shown in Figure 4.1, where we perform text-based texture editing using the method from[48] and subsequently render the head avatar in novel poses and views.

Similarly, we can control the shape of the underlying FLAME head model by changing its shape parameters β to achieve realistic-looking shape control over the Gaussian head avatar. See Figure 4.9 for shape-based control results, along with the learned normals.

4.5.2 Quantitative Comparison

Given that most learning-based monocular head avatar techniques assume uniformly lit environments with fixed illumination and are not designed to learn from polarization data, we limit our comparative analysis of training methodologies to existing methods using only the cross-polarized sequences from our captured dataset. We compare our approach against state-of-the-art (SOTA) methods, specifically PointAvatar[95] and GaussianBlendshapes[47] in Figure 4.6. As shown, our method outperforms both in terms of rendering quality for novel views/poses and in preserving fine details. Both PointAvatar and GaussianBlendshapes exhibit blurry and noisy details (highlighted in red) on novel frames, which can be attributed to the multiview inconsistency issues associated with 3DGS[39]. In contrast, our method produces noise-free, high-quality renderings. PointAvatar also demonstrates a high susceptibility to tracking noise, leading to inaccuracies such as incorrect eye rendering in (c) Figure 4.6. Furthermore, we conduct a quantitative evaluation on the same subjects (a), (b), and (c) and present these results in Table 4.1, where our method surpasses or is at par with other approaches in the majority of cases.

Our evaluations on a synthetic dataset, as presented in Figure 4.12 and Table 4.3, indicate that our method surpasses [95] in reconstructing surface normals. This is because [95] tends to generate overly smooth normals, which leads to a deficiency in capturing fine surface details and thus lacks fidelity.

Although our approach is primarily designed for reconstructing head avatars from polarized sequences, it possesses the flexibility to be extended to monocular, unpolarized (diffuse-only) RGB sequences. This adaptation involves deactivating the physically-based lighting model and representing appearance solely through the albedo map and the pose-dependent delta map. As clearly shown in Table 4.2, even under these conditions, our method outperforms [95].

Our proposed 2DGS-based representation facilitates the use of normal regularization constraints, leading to superior surface reconstruction quality compared to other state-of-the-art (SOTA) methods that rely on 3DGS. A qualitative comparison of surface normal quality is presented in Figure Figure 4.11. Our approach demonstrates higher quality renderings and significantly more detailed surface normals when compared to PointAvatar [95] and GaussianBlendshapes[47]. It is worth noting that although [95] incorporates the Eikonal constraint to regularize surface normals, its inherent SDF-based representation

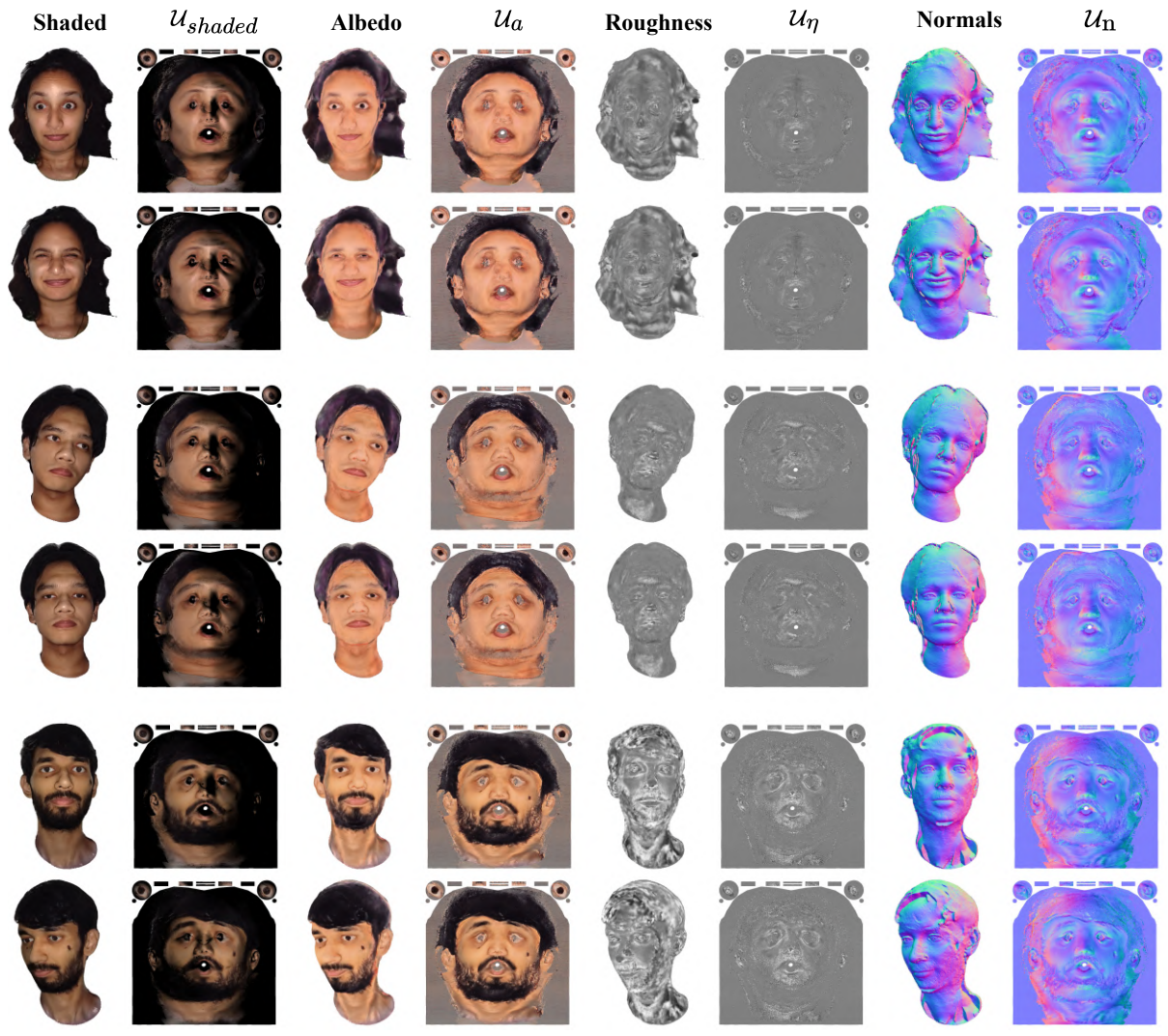


Figure 4.10 Additional Qualitative Results

Table 4.1 Quantitative evaluation on subjects (a), (b) & (c).

Methods	Metrics	(a)	(b)	(c)
Pt.Av[95]	L_1(1e-2)↓	1.33	1.31	1.11
	PSNR↑	27.90	27.03	28.38
	SSIM↑	0.92	0.94	0.94
GBS[47]	L_1(1e-2)↓	1.15	2.14	**0.84**
	PSNR↑	28.95	25.86	**31.60**
	SSIM↑	0.91	0.85	0.94
Ours	L_1(1e-2)↓	**1.03**	**0.67**	0.93
	PSNR↑	**29.79**	**31.28**	30.28
	SSIM↑	**0.94**	**0.96**	**0.95**

Figure 4.11 Surface Normal Rendering Comparison with PointAvatar[95] and Gaussian Blendshapes[47].

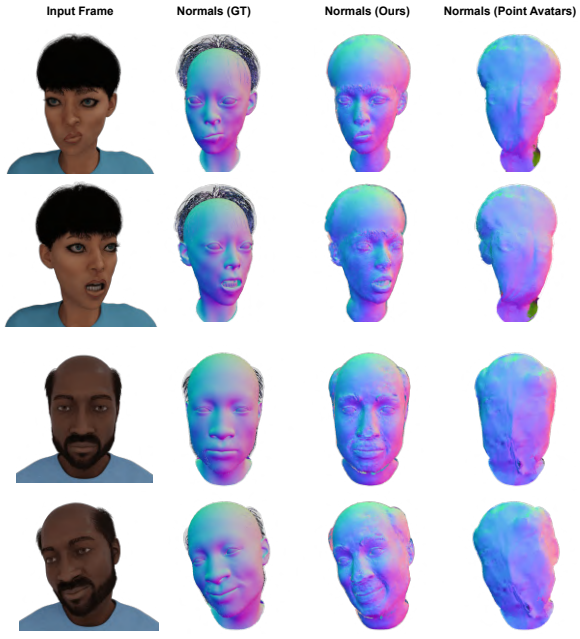


Figure 4.12 Surface Normal Comparisons on Synthetic Data.

Methods	Metrics	Subject-1	Subject2
Pt.Av	$L_1(1e\text{-}2)\downarrow$	0.97	1.30
	PSNR↑	28.71	29.53
	SSIM↑	0.921	**0.931**
Ours	$L_1(1e\text{-}2)\downarrow$	**0.65**	**0.69**
	PSNR↑	**33.99**	**34.64**
	SSIM↑	**0.923**	0.920

Table 4.2 Quantitative comparison of different methods. ↑: higher is better, ↓: lower is better. The best results are in **bold**.

tends to produce overly smooth normals when performing monocular reconstruction with limited multi-view information.

Figure 4.11 displays rendered surface normals alongside the corresponding rendered RGB images for our method, as well as for [95] and [47]. Method [47] generates noisy surface normals because normals are not well-defined for 3D Gaussian Splats, and it does not apply any regularization to them. (To render these surface normals, we define the normal axis as the splat's axis with the minimum span.) While the surface normals produced by [95] are better, they are excessively smooth due to the low-frequency bias of the MLP used in their approach. Furthermore, the RGB renderings from [95] also lack the high-frequency details that are present in the results from [47] and our method.

Figure 4.12 offers a comparison between our method and [95] using a synthetic dataset, clearly illustrating the superior surface normal reconstruction capabilities of our approach. This synthetic dataset was created on a per-subject basis using Blender.

4.5.3 Ablation Study

We conducted an ablative analysis to assess the impact of several modelling choices integrated into our training procedure, with a summary of the qualitative results presented in Figure 4.13.

Bilateral Grid $\mathcal{U}_W$: As shown in the figure, the bilateral grid $\mathcal{U}_W$ is critical during the second training stage. It functions by adjusting the tint of the albedo map (which was initially learned from cross-polarized images) to accurately match the tint observed in the parallel-polarized images.

Pose-conditioned MLP Φ_Δ: The results indicate that the pose-conditioned MLP, Φ_Δ, plays a significant role in preserving pose-specific fine details, such as wrinkles, when rendering novel poses.

FLAME-based regularization $L_{\textbf{flame}}$: Empirical observations show that the L_{flame} term in Equation 4.12 effectively regularizes noise within the learned UV maps. It achieves this by compelling

Methods	Metrics	Subject 1	Subject 2
Ours	PSNR↑	**17.92**	**20.83**
	MSE↓	**1.59e-2**	**8.19e-3**
	L_1 ↓	**0.0512**	**0.0421**
Point-avatar	PSNR↑	16.84	19.22
	MSE↓	$2.12e-2$	$1.23e-2$
	L_1 ↓	0.0647	0.0558

Table 4.3 Comparison between normal rendering of Our method and [95] on synthetic data. ↑: higher is better, ↓: lower is better. The best results are in **bold**.

Table 4.4 Quantitative ablation on DuoPolo subjects

Config.	L_1↓	SSIM↑	PSNR↑
Ours w/o Φ_Δ	0.801	0.812	30.07
Ours w/o $\mathcal{U}_\mathrm{W}$	2.303	0.928	23.76
Ours w/o $\mathcal{L}_{flame}$	0.811	0.948	30.25
Ours	0.848	0.944	30.67

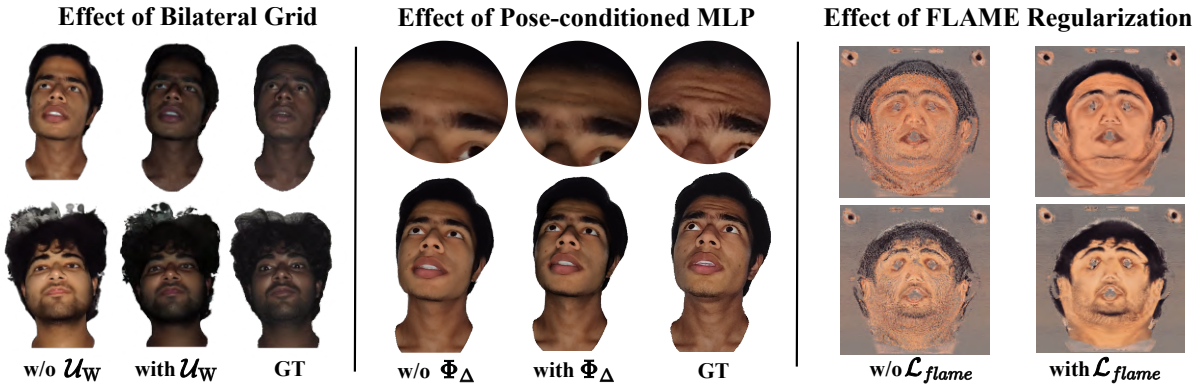


Figure 4.13 Qualitative analysis of different training configurations.

unobserved texels, particularly in the initial training iterations, to align with the rendering of the base FLAME mesh.

4.5.4 Training & Implementation Details

Texture maps of size 1024×1024 are learned using Hash-encoded Coordinate MLPs. For texture map synthesis, the Hash-encoded MLP employs 1 hidden layer with 16 neurons. Its associated Hash Table features 7 levels, each of size 2^{20}, with a base resolution of 512 and a growth factor of 1.26. The separate pose-dependent MLP is designed with 2 hidden layers, each containing 32 neurons. The Hash grid for this pose-dependent MLP consists of 16 levels, each with a size of 2^{18}, a base resolution of 16, and a growth factor of 1.4. Both MLPs are trained using the AdamW [45] optimizer with a learning rate of 1×10^{-3}. To reconcile the albedo map between cross-polarized and parallel-polarized streams, we utilize a bilateral grid. This learnable bilateral grid has dimensions of $16 \times 16 \times 8 \times 12$ and is also optimized with AdamW using a learning rate of 1×10^{-3}. Training for each sequence involves 16,000 iterations in the first stage and 10,000 iterations in the second stage. All experiments were executed on an NVIDIA RTX A6000 GPU.

4.6 Discussion & Limitations

Our proposed work offers the potential to democratize access to relightable head avatars by making polarized facial performance acquisition significantly more accessible and cost-effective. The monocular polarized dataset created using our proposed capture approach is unique. However, its quality is not directly comparable to data from advanced volumetric lightstage systems. Smartphone camera limitations constrain our dataset's resolution, occasionally causing inaccuracies in modelling fine-grained specular reflections or details like hair strands. Furthermore, the restricted range of lighting variations in our capture makes generalizing relighting to complex environment maps challenging. Since the angle between viewing direction and light direction always remains zero, the effect of light at grazing angles becomes difficult to model.

Owing to our capture setup, which consists of a single point light source in a dark room, objects that are dark become difficult to model, like black hair. The normals in those regions become ill-defined because multiple solutions can exist for those regions during optimisation.

Additional constraints stem from the underlying parametric head model, which struggles with the accurate representation of intricate inner-mouth areas like the tongue and teeth, sometimes leading to blurry details in these regions. The 2DGS representation also has difficulty capturing very fine details such as individual beard hairs, hair strands, or eyelashes. To overcome these issues, future work will involve investigating neural strand-based representations for more precise modelling of these facial attributes. We also plan to explore more accurate differentiable rendering techniques, like differentiable ray tracing, as an alternative to Gaussian splatting, to better model physics-based reflectance properties.

The proposed approach is very susceptible to noise while tracking the FLAME head model, and might product noisy expression conditioned albedo maps as a result. The expression-dependent MLP at the time of inference proves to be a bottleneck while rendering the model, limiting the rendering FPS to around 40 frames per second. A single head model takes about 90 minutes to train using the pro-

vided approach, which might not be acceptable while creating a large dataset. Moreover, the presented approach could not support environment map based relighting.

Since our work generates a photoreal avatar that can be used to reenact any actor, it might raise some ethical concerns. We also collected a dataset involving facial information of 10 subjects. We are fully aware of the potential privacy concerns inherent in such data and have proactively implemented responsible measures to address them. All participation was entirely voluntary, with explicit informed consent obtained from every subject before their inclusion. Each individual was thoroughly briefed on the dataset's scope, purpose, and potential applications to ensure full transparency and understanding. To further protect privacy and minimize risks, we have established strict data access protocols. Access is granted only after a comprehensive review and approval process, which requires researchers to submit a formal application and commit to ethical guidelines. We stress that this dataset is intended exclusively for academic and research use, and its distribution is carefully controlled to prevent misuse.

While our method aims to advance research in relightable head avatars, we acknowledge the potential for unethical misuse, such as the creation of deepfakes. Although deepfakes can have legitimate uses in entertainment and education, they are often linked to harmful outcomes like misinformation, identity theft, and digital harassment. Our method cannot create a head avatar out of an arbitrary identity, it needs to be consensually captured first. However, we believe it is crucial to address the ethical considerations of methods like ours to responsibly balance technological advancement with societal impact. By promoting transparency, accountability, and oversight, and by strictly limiting the use of our dataset and methodology to research purposes, we strive to minimize potential risks while fostering the legitimate and responsible application of our work.

4.6.1 Summary

We proposed a novel 2DGS-based head representation, along with a novel smartphone-based setup, which marks a significant step towards making relightable head avatars scalable. However, the efficiency of the proposed approach remains questionable due to its long training times, as relatively slow rendering fps. In the following chapter, we will improve upon the proposed head avatar, both in terms of quality of representation and in terms of computational efficiency.

Chapter 5

LightHeaded++: Improved Relightable & Editable Head Avatars from a Smartphone

In this chapter we will discuss an improved formulation for the head avatar representation that was proposed in the previous chapter. As discussed earlier, the proposed representation suffered from several limitations, like long training times, relatively low rendering and limited support for relighting. It did not outperform existing methods in a monocular setup, which is the most widely available capture method.

We therefore proposed an improvement in the framework that enables training a Gaussian head avatar in under 10 minutes, with real-time rendering speeds and a low memory footprint. The new and improved framework is less susceptible to noise in tracking and surpasses previous state-of-the-art methods across all datasets. We will highlight the improvements in the method and re-evaluate our method across a wider range of existing methods, and prove the efficacy of our proposed framework.

5.1 Improved Framework

Compared to the previous approach, our polarizer based capture setup remains the same, and the head is represented as 2DGS embedded onto a parametric FLAME head model. The major improvements lie in the fact that expression-dependent residual maps are estimated, as well as improvements in the shader model to support environment-based lighting. We no longer require an MLP at the inference step, providing a considerable speed-up in rendering speeds. The changes made to the framework are described ahead:

5.1.1 Textured Gaussian Head Representation

Similar to our previous representation, we obtain a 2DGS based head avatar which enables us to get accurate surface normals. In our improved framework as well, the gaussian primitives are learnt over the UV space of FLAME. However, the way attributes are embedded in the UV space is modified.

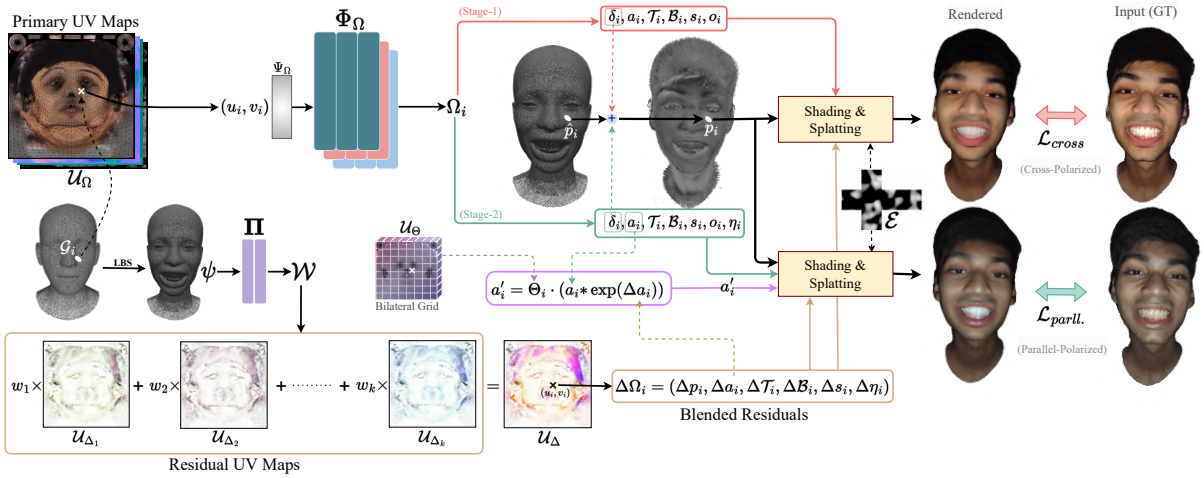


Figure 5.1 Proposed two-stage training strategy to learn textured Gaussian head avatars with decomposed appearance and geometry.

5.1.2 Embedding 2D Gaussians in UV space:

Similar to the original framework, our process commences by initializing a 2D Gaussian, G_i, for each UV coordinate (u_i, v_i) within FLAME's canonical UV space $\mathcal{U}$. We also perform a one-time computation to estimate a face index map, $\mathcal{U}_{idx}$, which stores the triangle (face) index 'm' corresponding to each UV coordinate in $\mathcal{U}$ (this index is undefined for empty UV regions).

To directly transform these Gaussians from their canonical pose/expression to the deformed pose/expression dictated by the posed mesh M, we first retrieve the associated face index $m = \mathcal{U}_{idx}(u_i, v_i)$. This provides us with the triangle $t_m = (v_{m0}, v_{m1}, v_{m2}) \in 3 \times \mathbb{R}^3$, representing the 3D positions of its vertices. For every Gaussian G_i, we then calculate its initial mean position, $\hat{p}_i$, in the posed 3D space using barycentric coordinates:

$$\hat{p}_i = (\alpha_i * v_{m0}) + (\beta_i * v_{m1}) + (\gamma_i * v_{m2}), \tag{5.1}$$

where $\alpha_i, \beta_i, \gamma_i$ are the barycentric coordinates for the point (u_i, v_i), and they sum to one ($\alpha_i + \beta_i + \gamma_i = 1$).

However, unlike the previous approach, where the Gaussians were free to move across the UV space, the coordinate of each Gaussian in the UV space is fixed and do not change over time. We empirically found that this change did not lead to a lowering of learning capability but helped in reducing the number of learnable parameters for each Gaussian. Instead of allowing freedom of movement in the 2D UV space, we allow more freedom in the 3D object space, where it can move in 3 spatial dimensions from its position on the mesh, conditioned on expression, instead of just allowing one degree of freedom along the normal direction.

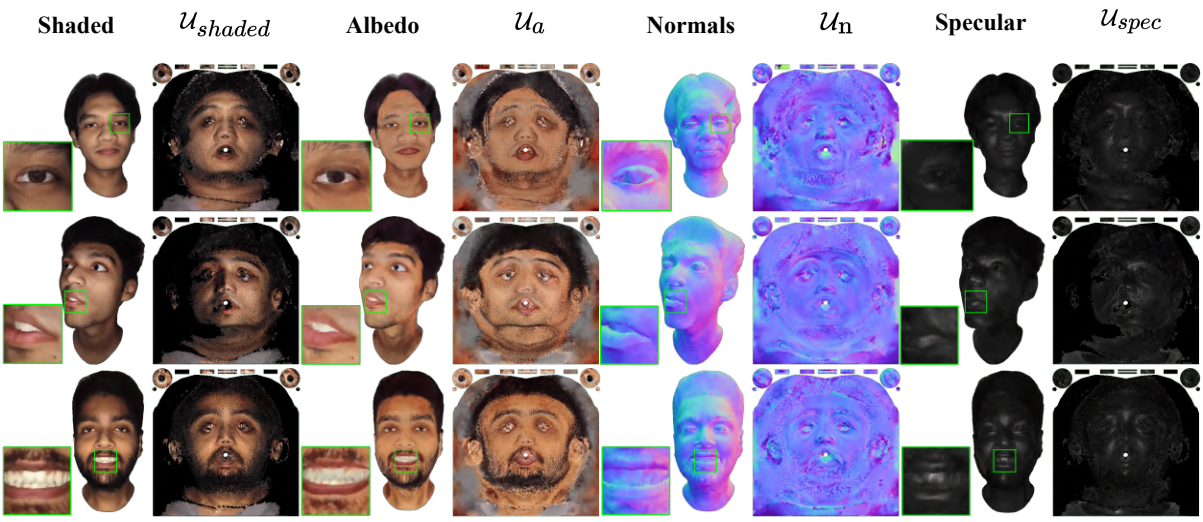


Figure 5.2 Decomposition of appearance & geometry in UV space.

5.1.3 Gaussian Attributes as UV Maps

Due to changes in the representation of our Gaussian head avatar, and how the gaussian are linked to the head, some modifications were necessary in defining the attributes of 2D Gaussians in 3D space.

Following the previously established formulation, we initiate by sampling 2D Gaussians with a defined a set of primary UV maps corresponding to each attribute: $\mathcal{U}_a$ (albedo map), $\mathcal{U}_\eta$ (roughness map), $\mathcal{U}_T$ (tangent map), $\mathcal{U}_B$ (bitangent map), $\mathcal{U}_\delta$ (offset map), $\mathcal{U}_s$ (scale map), and $\mathcal{U}_o$ (opacity map). Similar to before, for any specific Gaussian G_i associated with its UV coordinate (u_i, v_i), we can query any of its attributes Ω_i using the corresponding map $\mathcal{U}_\Omega(u_i, v_i)$. The attribute Ω_i can represent its albedo color $a_i \in \mathbb{R}^2$, roughness $\eta_i \in \mathbb{R}$, tangent vector $T_i \in \mathbb{R}^3$, bitangent vector $B_i \in \mathbb{R}^3$, 2D scale $s_i \in \mathbb{R}^2$, or opacity $o_i \in \mathbb{R}$.

To estimate the orientation for a 2D Gaussian G_i, we first retrieve its tangent $T_i = \mathcal{U}_T(u_i, v_i)$ and bitangent $B_i = \mathcal{U}_B(u_i, v_i)$ from their respective UV maps and then apply Gram–Schmidt orthogonalization. The tangent space rotation for this Gaussian G_i is subsequently defined as $\hat{R}_i = [T_i, B_i, T_i \times B_i] \in \mathbb{R}^{3\times3}$. We then derive the 3D world space rotation/orientation R_i by transforming $\hat{R}_i$ with a matrix T ($R_i = T * \hat{R}_i$), where T is a 3×3 matrix that converts vectors from tangent space to world space. Additionally, we compute the normal vector $N_i = T_i \times B_i$, which is essential for the shading process.

To account for geometric features that are distant from the surface of the base parametric head mesh M, we define an offset $\delta_i = \mathcal{U}_\delta(u_i, v_i)$ along the normal direction. This offset is then used to shift the initial mean position $\hat{p}_i$ of the 2D Gaussian within the 3D space, yielding the final position p_i as follows:

$$p_i = \hat{p}_i + \xi_i * T * \delta_i \tag{5.2}$$

Here, ξ_i is the area-adjustment factor, same as before :

$$\xi_i = (\alpha_i * a_{m0}) + (\beta_i * a_{m1}) + (\gamma_i * a_{m2}) \tag{5.3}$$

where

$$a_{mj} = \frac{1}{|N(v_{mj})|} \sum_{k \in N(v_{mj})} \sqrt{\text{Area}(t_k)} \tag{5.4}$$

In these equations, α, β, γ represent barycentric coordinates, and a_{ij} (for vertex v_{ij}) is the mean root-sum of the areas of triangles within the neighbourhood $N(v_{mj})$ of vertex v_{mj}.

5.1.4 Residual UV Maps:

The majority of our changes in the new approach lie in the way the expression-dependent residual maps are learnt. Earlier, the residual map was applied only to the albedo map $\mathcal{U}_a$. As a result, it could only model appearance changes due to expression changes, but no expression-dependent geometry changes could be modelled, beyond what was possible by LBS of the FLAME head mesh. Therefore, in the new framework, we learn residual maps for all other attributes, which enables us to achieve very high frequency appearance and geometry changes due to expression.

Beyond the primary UV maps, we also manage a collection of UV maps dedicated to storing expression-dependent residuals. We establish 'k' such residual UV maps. For each (u_i, v_i) coordinate, these maps store the residuals $\Delta\Omega_i$, which are applied on top of the primary attributes Ω_i. This allows us to model the various geometrical and appearance alterations observed across the video sequence. Specifically, the residuals we store are $\Delta p_i, \Delta a_i, \Delta T_i, \Delta B_i, \Delta s_i$, and $\Delta \eta_i$.

To incorporate expression guidance, we take the expression parameters of the FLAME head mesh, $\psi \in \mathbb{R}^{100}$, and project them onto a k-dimensional space. This projection is achieved using a projection matrix, $\Pi \in \mathbb{R}^{k \times 100}$, which yields linear blend weights $W = \Pi \cdot \psi \in \mathbb{R}^k$. The final blended residuals for a specific Gaussian G_i are then computed as follows:

$$\Delta\Omega_i = \mathcal{U}_\Delta(u_i, v_i) = \sum_{w_j \in W} w_j * \mathcal{U}_{\Delta j}(u_i, v_i) \tag{5.5}$$

These calculated residuals are subsequently added to the primary attributes Ω_i to derive the final attributes Ω'_i, according to the following equation:

$$\Omega'_i = \begin{cases} \Omega_i + \Delta\Omega_i & ;\text{for } p_i, T_i, B_i \\ \Omega_i * \exp(\Delta\Omega_i) & ;\text{for } a_i, s_i, \eta_i \\ \Omega_i & ;\text{for } o_i. \end{cases} \tag{5.6}$$

Since all the residual maps can be represented as k residual maps, the need for an MLP-based residual map generator is eliminated. This allows us to have an inference which is free of any feed-forward neural network, and each residual map generation, which was a bottleneck earlier, becomes a simple linear blending of basis residual expression maps as seen in Figure 5.1

5.1.5 Rendering Equations:

The Gaussians are still represented as an RGB colour c_i where view-dependent effects arise due to the shading model. The shading model for point light remains similar, where we use a physically based SVBRDF shading model, with Cook-Torrence [16] microfacet specular BRDF

For the diffuse component, we continue to use the BRDF model proposed by Ashikhmin & Shirley[3].

Where our approach differs is the fact that we now model an ambient shading component f_{env}, which is due to light bouncing off the environment surfaces. This component becomes essential because the flashlight source is not a perfect point light source, and our initial assumption assumed no secondary bounces. We assume that optimizing an environment map along with the point light-based assumption would help us capture these secondary bounces of light. This also helps us better relight the head model using an environment map at the time of inference.

We follow the differentiable version of the split sum shading model proposed in [54] to learn environment lighting from image observations through optimization. The final shaded color is then computed as:

$$c_i = f_{di} + f_{si} + f_{\text{env}} \tag{5.7}$$

Figure 5.1 illustrates our new training methodology for learning a textured Gaussian head avatar. Given input cross-polarized and parallel-polarized video sequences, we learn the aforementioned Gaussian attributes in a two-stage process. We employ a set of stacked Multi-Layer Perceptrons (MLPs), denoted Φ_Ω, which consists of several hash-encoded MLPs [52] using hash-encoding Ψ_Ω. These MLPs predict UV attributes $\Omega_i = \mathcal{U}_\Omega(u_i, v_i)$ for each 2D Gaussian G_i. For residual UV maps, we initialize learnable UV tensors $\mathcal{U}_{\Delta j}$ with $z = |\Delta\Omega_i|$ channels, where j ranges from 1 to k. To address the global tint-shift between the cross-polarized and parallel-polarized frames, we use a low-dimensional Bilateral Grid [85] as before, $\mathcal{U}_\Theta$, with a learnable affine transformation matrix Θ_i for each UV texel. Additionally, because we allow the (u_i, v_i) coordinates to optimize over time, we precompute the triangle-index map $\mathcal{U}_{idx}$ for efficient querying of triangle indices from UV coordinates. Finally, we define a learnable cube-map E to store environment lighting information. All values specified within $\mathcal{U}_\Omega$, $\mathcal{U}_{\Delta j}$, $\mathcal{U}_\Theta$, Π (projection matrix for blend weights), and E are initialized randomly and optimized via differential rendering losses in a self-supervised manner. This uses the cross-polarized sequence in the first stage and the parallel-polarized sequence in the second.

During the first stage, in this framework, our focus is on learning $\mathcal{U}_\Omega$ (excluding the roughness map $\mathcal{U}_\eta$) along with the residual UV maps $\mathcal{U}_{\Delta j}$. We input (u_i, v_i) into the hash-encoded MLP Φ_Ω to predict attributes $\Omega_i = \Phi_\Omega(\Psi_\Omega(u_i, v_i))$. This includes the albedo a_i, tangent T_i, bitangent B_i, scale s_i, opacity o_i, and an estimate of the shifted mean position p_i (as per Equation 5.2). For a given pose/expression θ, we compute linear blending weights $\psi = \Pi \cdot \theta$ and use Equation 5.5 to obtain the residuals $\Delta\Omega_i$. These residuals are then added to the primary attributes using Equation 5.6. Finally, we compute the final shaded color c_i via Equation 5.7 (ignoring the specular component at this stage) and use this color,

along with T_i, B_i, s_i, and o_i, to perform 2D Gaussian splatting for rendering the image and minimizing the loss L_{cross}.

In the second stage of training, we fix the primary and residual albedo UV maps ($\mathcal{U}_a$ and $\mathcal{U}_{\Delta a}$) by freezing their optimization. Our focus shifts to learning the roughness map exclusively. Additionally, we learn the bilateral grid $\mathcal{U}_\Theta$ during this stage. This is to account for the tint shift observed between the albedo learned from cross-polarized images in the preceding stage and the appearance in the parallel-polarized images. The tint-corrected albedo is then obtained as:

$$a'_i = \Theta_i \cdot (a_i * \exp(\Delta a_i)). \tag{5.8}$$

During this second stage, when computing the shaded color c_i (using Equation 5.7), we incorporate both diffuse and specular components. We retrieve the values of the other attributes learned in the first stage, render an image via splatting, and then minimize the loss L_{parll}. Both L_{cross} (from the first stage) and L_{parll} are defined by the following loss composition:

$$L_{\text{cross/parall.}} = L_1 + L_{\text{SSIM}} + L_{\text{LPIPS}} + L_{\text{scale}} \tag{5.9}$$

where $L_{\text{scale}} = \sum_i \max(0, s_i - 0.3)$ remains the same

Note that the L_{flame} term has been removed, as we empirically found it to not affect the visual quality much, but was computationally inefficient, as for each training step, besides the gaussian based rendering, it also became necessary to do a mesh rasterization based rendering of the FLAME head. Removing this term helped us to considerably improve the training times.

Throughout both training stages, we initialize the environment cube map E randomly and optimize its values using Equation Equation 5.7. For efficient rendering during inference, we discard the MLP Φ_Ω. Instead, we directly use the attributes stored in the primary UV maps $\mathcal{U}_\Omega$, while continuing to rely on the projection matrix Π and the residual UV maps $\mathcal{U}_\Delta$ to handle deformations dependent on poses and expressions.

It is important to note that even though we use Hash-grid MLP to learn the primary map, as seen in Figure Figure 5.1, we can discard the MLP once the primary maps are learnt; therefore, at inference, the need for Hash-grid MLP is eliminated for both the primary texture attribute maps and the residual maps. The residual maps are like a codebook of parameters, which are optimized directly without the need for an MLP.

5.2 Training & Implementation

Our new approach learns texture maps at a resolution of 512×512 pixels using Hash-encoded Coordinate MLPs. These Hash-encoded MLP responsible for texture map synthesis are designed with 2 hidden layer containing 128 neurons each. Its associated Hash Table is structured with 16 levels, where each level has a size of 2^{22}. The base resolution for this hash table is 8, and it employs a growth factor of 1.26.

For pose, we create a codebook containing residual UV Map basis, which can be linearly combined to produce a residual texture map for the various Gaussian attributes. The Flame expression parameters ($\psi \in \mathbb{R}^{100}$) are mapped to a lower dimensional space using a learnt linear transformation $\Pi \in \mathbb{R}^{k \times 100}$ to get the linear weights $\mathcal{W} \in \mathbb{R}^k$ which are used to weigh the basis maps. The value of $k = 25$. Therefore, the total size of the codebook becomes $25 \times 512 \times 512 \times 17$, where the last dimension represents the various residual attributes that need to be learnt. This set of parameters is also optimized via Adamw [45] optimizer with a learning rate of 10^{-3}.

To ensure color consistency by matching the albedo map between the cross-polarized and parallel-polarized video streams, we employ a bilateral grid. This learnable bilateral grid has dimensions of $16 \times 16 \times 8 \times 12$.

We maintain the bilateral grid to ensure colour-consistency between the cross-polarized and parallel-polarized streams. This learnable bilateral grid has dimensions of $16 \times 16 \times 8 \times 12$ and is also optimized using the AdamW optimizer with an identical learning rate of 1×10^{-3}.

The training process for each sequence is divided into two stages: the first stage runs for 15,000 iterations, and the second stage runs for 5,000 iterations. All computational experiments were conducted on an NVIDIA RTX A6000 GPU. Two-stage training takes about 20 minutes, while single-stage training, on the monocular dataset, can be completed in as little as 6 minutes.

5.3 Experiments & Evaluation

Evaluation Dataset & Metrics: Our experimental evaluations are conducted using sequences from the "DuoPolo"section 4.4 dataset, which we introduced in previous chapter. However, unlike the previous framework, we also compare on two other datasets [1][47], which proves the efficacy of our method across various identity types and input video resolutions. For every sequence, we implement a train-test split, allocating the initial 80% of its frames for training purposes and reserving the final 20% for testing. In our quantitative assessment, we calculate the same metrics as before, such as the per-sequence (or alternatively, per-subject) L_1 error, Peak Signal-to-Noise Ratio (PSNR)[19], and Structural Similarity Index Measure (SSIM)[32]. These metrics are computed by comparing the frames rendered by our model against the corresponding input (ground truth) frames taken from the designated test portion of the data.

5.3.1 Qualitative Results

Learned Reflectance Maps for Relighting: The capability of our proposed method to capture physically plausible facial reflectance is demonstrated in Figure 5.2. This is achieved by successfully separating appearance into distinct albedo, roughness, and normal components, represented as UV maps. Concurrently, our method reconstructs high-frequency geometrical details. This effective disentanglement

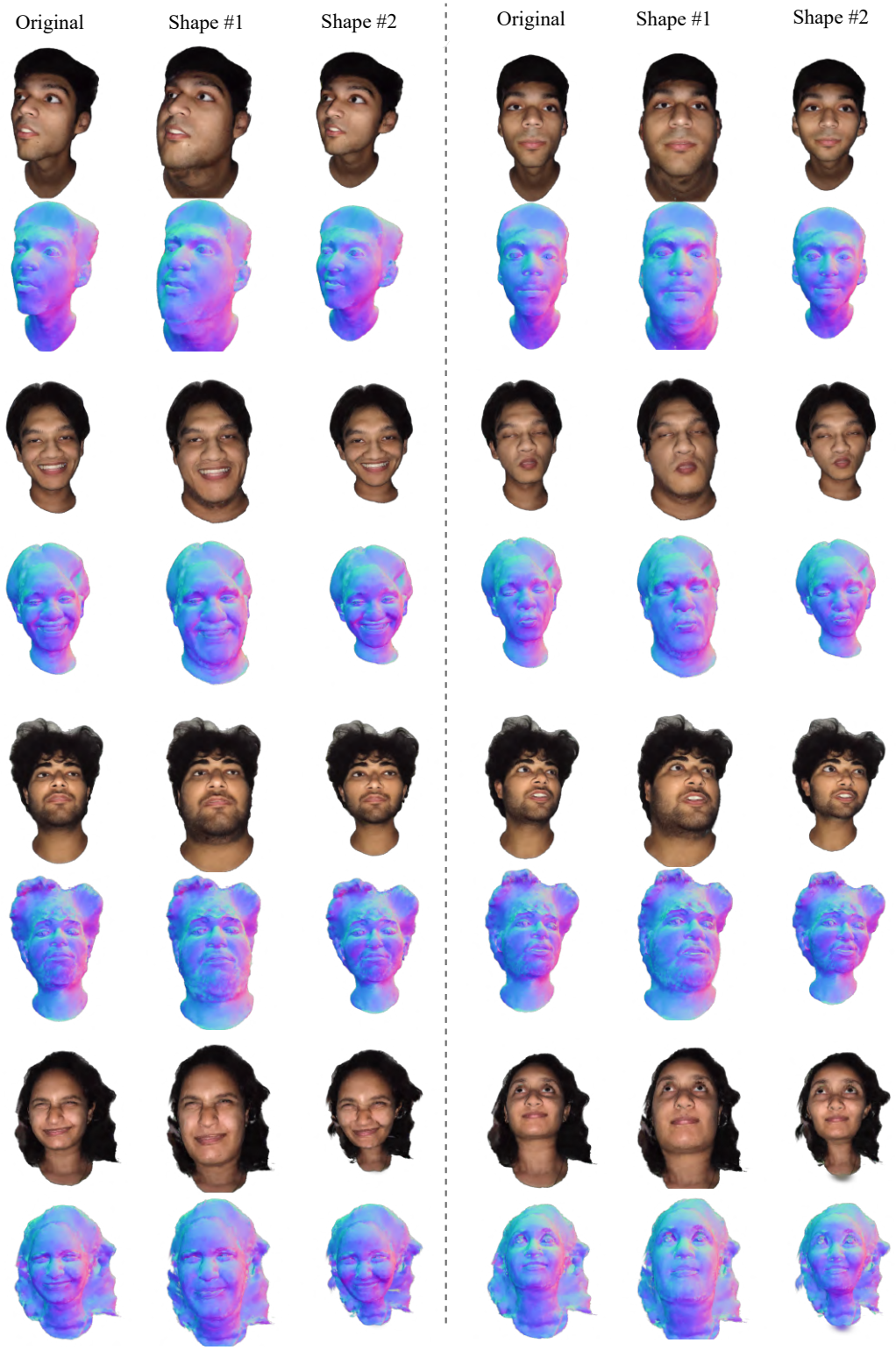


Figure 5.3 Shape editing over reconstructed head avatars.

Table 5.1 Comparison of different methods on INSTA, 3DGB and Our Dataset. Best values are highlighted in green, while second-best values are in yellow.

	INSTA[1]			**3DGB [47]**			**DuoPolo (Ours))**		
Method	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Point-Avatar	29.12	0.932	0.094	31.76	0.926	0.145	27.77	0.919	0.128
FlashAvatar	30.14	0.942	0.038	25.92	0.920	0.094	29.79	0.923	0.089
Gaussian Deja-vu	25.56	0.925	0.058	23.45	0.910	0.079	24.73	0.848	0.183
SPARK	23.75	0.873	0.103	24.70	0.863	0.101	23.75	0.873	0.103
GaussianBlendshapes	30.01	0.951	0.084	32.05	0.942	0.144	30.02	0.943	0.094
Ours	30.33	0.952	0.036	32.92	0.943	0.046	31.98	0.955	0.053

enables the relighting of the reconstructed head avatars using varied environment maps, as illustrated in Figure Figure 5.4.

Editing: Our proposed textured human head representation still supports both shape and appearance editing. Shape modifications can be made by altering the shape parameters of the underlying FLAME mesh (shown in Figure 5.3). Appearance editing is facilitated by directly modifying only the albedo map like before.

5.3.2 Comparison

Given that the majority of learning-based monocular head avatar methods operate under the assumption of a uniformly lit environment with constant illumination and are not designed to process polarization data, we conduct a fair comparison of our training methodology against existing state-of-the-art (SOTA) methods by focusing exclusively on the cross-polarized (diffused) sequences from our own captured dataset. To broaden this comparison, we also incorporate samples from the INSTA [1] dataset and the 3DGB dataset introduced in [47]. We adhere to the comprehensive evaluation strategy outlined by GaussianBlendshapes (GBS)[47] and present our results for PSNR, LPIPS, and SSIM in Table 5.1. These results demonstrate that our method outperforms all the existing methods for head avatar reconstruction from monocular video.

A qualitative comparison, presented in Figure Figure 5.6, demonstrates the superior quality of our results when compared to other methods. Our approach successfully captures fine details (as highlighted within boxes in the figure) and, importantly, also supports relighting and editing capabilities, features not directly offered by the compared techniques.

Furthermore, we compare the surface normals of the head avatar generated by our method against those from approaches that aim to reconstruct facial geometry in addition to appearance, as shown in Figure 5.5. The results indicate that PointAvatar produces overly smooth geometry because its point

Figure 5.4 Relighting the reconstructed head avatars in diverse environments.

splats are unable to represent detailed geometric features. In contrast, while [95] GBS [47] attempts to capture more detail, it generates numerous spiky artefacts, primarily due to the inherent limitations of 3d Gaussians, which are not ideally suited for accurate surface reconstruction. SPARK[6] models facial geometry as a mesh by learning offsets on top of an underlying FLAME head mesh; however, it struggles to capture high-frequency details such as wrinkles, beards, or loose skin. Conversely, our proposed 2DGS-based head representation yields high-quality surface normals that effectively capture subject-specific finer details.

5.3.3 Ablation Study

Due to changes in our formulation in the new framework, we conducted new ablative analysis to evaluate the contributions of several key components proposed in our method. The impact of employing residual UV maps, $\Delta\Omega$, is demonstrated in Table 5.2. Furthermore, we investigate the effect of varying the dimensionality (the value of 'k') of the low-dimensional projection space used for linear basis blend weights, with results presented in Table 5.3. Our observations indicate that selecting a higher value for 'k' enhances the quality of the results; however, this also leads to a significant increase in storage

Figure 5.5 Comparison for surface normals of the reconstructed facial geometry.

requirements. A larger 'k' value also necessitates more learning parameters, consequently extending the training time. Additionally, we examine the influence of UV map resolution in Table 5.4, illustrating the trade-off between rendering quality and training duration. While higher resolution maps utilize more Gaussians and thus result in slower convergence, the corresponding improvements in rendering quality are not substantial. Therefore, we adopt 512×512 as our default UV map resolution.

Table 5.2 Effect of Residual Maps

Texture Maps	**PSNR** ↑	**SSIM**↑	**LPIPS**↓
w $\Delta\Omega_i$	**33.12**	**0.953**	**0.048**
w/ o $\Delta\Omega_i$	30.99	0.945	0.053

5.4 Discussion

Our new framework outperforms the previous approach not only on our dataset but on all datasets, unlike before. The new approach has faster training time, high representative ability, produces higher quality head avatar and adds support for environment map based lighting.

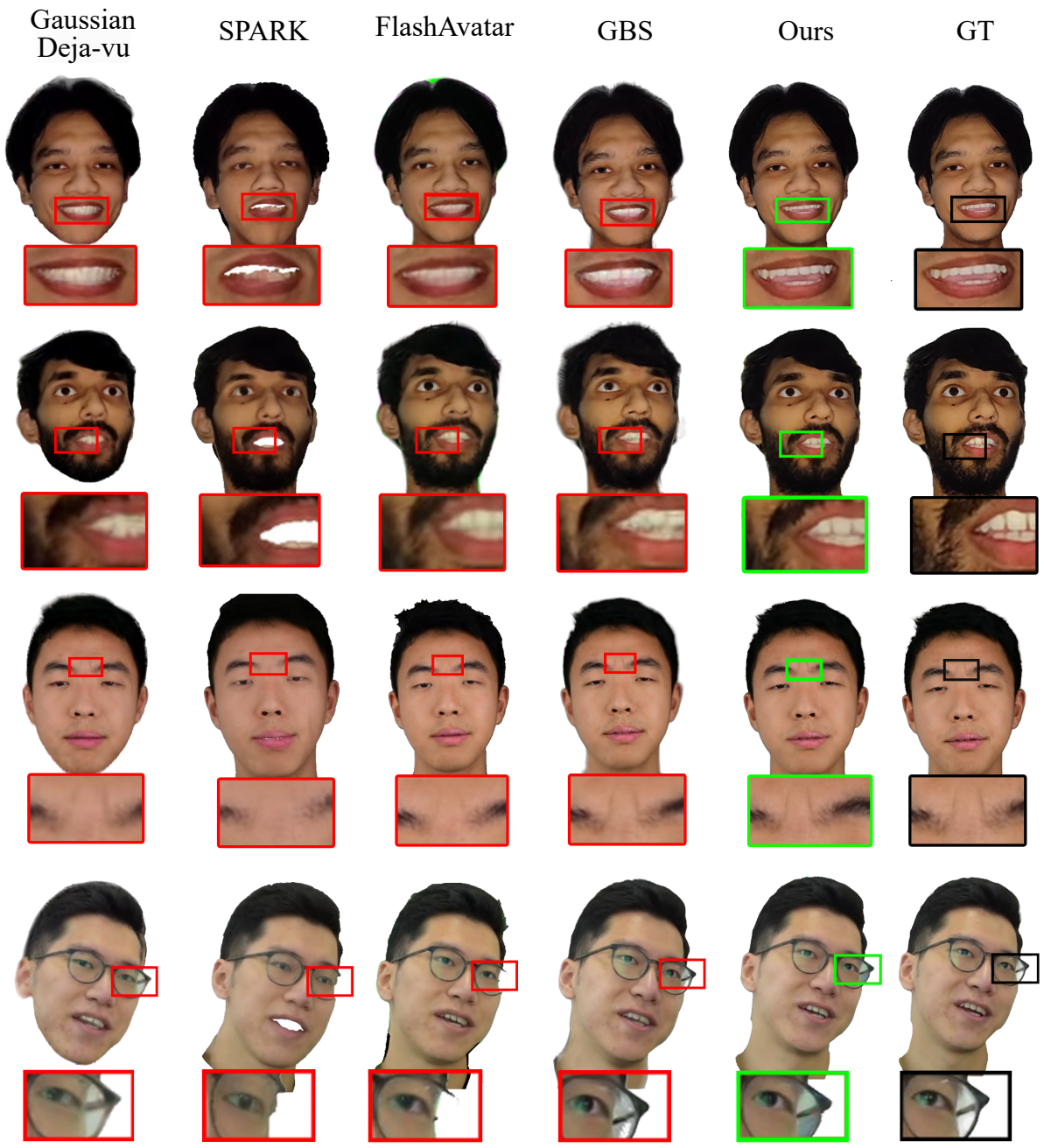


Figure 5.6 Qualitative comparison with SOTA methods. Our method is able to better capture finer details (such as teeth and eyes) whilst supporting relighting, pose and expression control.

Due to our novel residual map basis codebook, we are able to reduce the effect of noise in FLAME parameter estimation. As a result, the rendered head avatar appears more temporally consistent compared to our original approach.

Table 5.3 Ablation over values of k for residual basis

Num. Basis Maps	PSNR↑	SSIM↑	LPIPS↓	Storage(MB)↓
5	32.46	0.953	0.039	**52.4**
25	**33.91**	**0.960**	**0.035**	262.2
50	33.62	**0.960**	0.036	524.3
100	33.59	**0.960**	0.037	1048

Table 5.4 Effect of TexMap resolution

Texture Resolution	PSNR↑	SSIM↑	LPIPS↓	Time(min)↓
128x128	33.12	0.953	0.048	5.78
256x256	33.65	0.957	0.039	6.2
512x512	33.84	0.959	0.036	11.2
1024x1024	33.92	0.961	0.033	28.8

However, it still suffers from some of the limitations of our previous approach. Hair strands are still a challenge to be modelled, and since the capture setup remains the same, the inability to capture the interaction of light at grazing viewing angles is still difficult. The variation in lighting conditions is restricted, making it challenging to generalise relighting for complex environment maps.

We aim to explore more accurate techniques for differentiable rendering, such as differentiable ray tracing, as opposed to Gaussian splatting, to better model physics-based reflectance properties, especially subsurface scattering of light, which occurs in the skin.

5.5 Summary

In the previous two chapters, we introduced LightHeadEd, a method designed to be both scalable and affordable for the capture and reconstruction of animatable, relightable, and editable head avatars. We were able to achieve it using a commodity smartphone equipped with polaroid filters. Our approach featured several key contributions: a seamless process for acquiring dynamic polarized facial performances; a novel textured Gaussian head representation where 2D Gaussian attributes are embedded within the UV space of a parametric head mesh; and an effective self-supervised training scheme to reconstruct these head avatars from monocular video input.

We presented the two representations of head avatars that were developed during the progress of this project. The initial approach achieved remarkable quality in our captured dataset, but suffered some drawbacks in preserving high-quality details in a monocular camera-based dataset. The temporal consistency was heavily reliant on the noise in the FLAME head tracking approach. In the improved representation, the focus was on making the head avatar more efficient in terms of rendering and training. Besides this, the new representation also achieved improved temporal consistency and better quantitative metrics across all tested datasets, making it a significant leap forward for democratizing the capture and reconstruction of human head avatars.

LightHeadEd facilitates various applications, including relighting and text-based appearance editing, and demonstrates superior performance in geometry, appearance, and physically accurate shading when compared to existing methods. While the precise modelling of very fine details such as individual hair strands and skin pores presents current limitations, our future work aims to address these. If the dataset is expanded to make it more extensive and diverse, it can aid in the development of a more generalizable universal head model.

Chapter 6

Conclusion

To summarize, this thesis presented and investigated two core methodologies for 3D head avatar creation: a text-driven approach and a capture-based method. These techniques collectively strive towards the goal of creating realistic digital replicas efficiently and scalably. In this chapter, we will further discuss how this work has contributed to the field, with potential impact, and possibilities for future research and extensions in avatar generation.

6.1 Impact

This thesis aims to advance the field of 3D Head Avatar Generation & Reconstruction, by providing scalable and efficient approaches for digital head replica creation. The proposed approaches can be of great use to the field of AR/VR for the application of telepresence and holoportation. It can also be useful in the VFX industry for the purpose of actor reenactment, body double replacement and appearance modifications like de-ageing. Our proposed capture method can make the creation of relightable head avatars scalable, making it possible to create a large-scale head avatar dataset, on which future foundational models could be trained. We aim to publicly release the captured dataset while outlining the steps to recreate the setup in the hopes of aiding the 3D computer vision community.

6.2 Discussion and Limitations

Although both of our presented approaches outperformed existing state-of-the-art methods both qualitatively and quantitatively, they suffer from some limitations, which can also open up venues for future improvement and research.

Our text-conditioned textured head avatar generation pipeline was capable of generating a diverse set of head avatars with a wide variety of possible appearances, owing to the latent-diffusion model used [66]. However, the generated texture maps have light baked into them, making the generated avatars look unrealistic. Besides, it failed to model pose-dependent appearance changes like wrinkles.

An approach to overcome this limitation could have been to train on a dataset, which has lighting information decoupled from its appearance information, to directly generate albedo, normal and specular maps conditioned on text. However, no such dataset existed prior to our work. Therefore, we set out to create a novel, scalable capture setup and head avatar representation that could create a relightable head avatar. The data collected from this capture setup could be used to train a generic head generation model conditioned on image or text.

Although our proposed method outperformed other State-of-the-art methods, it suffered from some drawbacks. One of the major ones is the inability to accurately model hair. Due to our point light-based lighting, it becomes difficult to capture highly detailed images of the hair region, especially for darker hair colours, resulting in the reconstructed head avatar lacking finer, strand-level details. This also inhibits the editing options for the hair region.

Using our method, the interaction of light at grazing viewing angles in not observed, and thus the interaction of skin under all possible light directions becomes difficult to model. A second capture device will immensely help with capturing these interactions.

6.3 Future Directions

This thesis opens up promising avenues for future research. We describe some of them as follows:

- **Universal Head Avatar:** The end-goal of our capture-based approach is to create a dataset that could be used to train a universal head avatar model. Such a model would be capable of generating a relightable head avatar conditioned on just text or a reference image. Our capture process can be scaled up to capture such a dataset.

- **Hair Strand Reconstruction:** A field that remains challenging to solve is that of hair strand-based reconstruction, which is very helpful when realistically simulating hair. Light interacts with hair differently as compared to skin, and the dynamics of hair are also inefficient to model. Our representation could be combined with a strand-based hair representation to make the hair regions more realistic.

- **Full Body Avatar:** Our proposed head avatar representation can be extended to a full body human avatar with minor modifications, allowing efficient and relightable representation for the complete human body.

- **Expanded Capture Setup:** Many of the limitations of our capture-based approach could be resolved if we use two phone cameras instead of one. However, one major challenge that arises is that of frame and light syncing between the two devices.

Publications

Thesis Publications

- **Pranav Manu**, Astitva Srivastava, Avinash Sharma; *CLIP-Head: Text-Guided Generation of Textured Neural Parametric 3D Head Models*; **SIGGRAPH ASIA'23, Technical Communications**.
- **Pranav Manu**, Astitva Srivastava, Amit Raj, Varun Jampani, Avinash Sharma, P.J. Narayanan; *LightHeadEd: Relightable & Editable Head Avatars from a Smartphone*; **arXiv (under review at ICCV'25)**.

Other Publications

- Astitva Srivastava, **Pranav Manu**, Amit Raj, Varun Jampani, Avinash Sharma; *WordRobe: Text-Guided Generation of Textured 3D Garments*; **ECCV'24**.
- Amogh Tiwari, **Pranav Manu**, Nakul Rathore, Astitva Srivastava, Avinash Sharma; *ConVol-E: Continuous Volumetric Embeddings for Human-Centric Dense Correspondence Estimation*; **CVPRw'23**

Bibliography

[1] *Instant Volumetric Head Avatars*, 2023.

[2] C. H. A. R. G. G. S. A. G. S. A. A. P. M. J. C. G. K. I. S. Alec Radford, Jong Wook Kim. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*. PMLR, 2021.

[3] M. Ashikhmin and P. Shirley. An anisotropic phong light reflection model. *Journal of Graphics Tools*, 5, 01 2001.

[4] Autodesk Inc. Maya. `https://www.autodesk.com/products/maya/overview`, Since 1998.

[5] D. Azinović, O. Maury, C. Hery, M. Nießner, and J. Thies. High-res facial appearance capture from polarized smartphone images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023.

[6] K. Baert, S. Bharadwaj, F. Castan, B. Maujean, M. Christie, V. Abrevaya, and A. Boukhayma. Spark: Self-supervised personalized real-time monocular face capture. 2024.

[7] 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 (SIGGRAPH '99)*, SIGGRAPH '99, pages 187–194. ACM Press/Addison-Wesley Publishing Co., 1999.

[8] 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*, SIGGRAPH '99, page 187–194, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[9] Blender Foundation. Blender - a free and open 3d creation suite. `https://www.blender.org/`, Since 2002.

[10] F. L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6):567–585, 1989.

[11] C. Cao, D. Bradley, K. Zhou, and T. Beeler. Real-time high-fidelity facial performance capture. *ACM Transactions on Graphics (TOG)*, 34:1 – 9, 2015.

[12] C. Cao, T. Simon, J. K. Kim, G. Schwartz, M. Zollhoefer, S.-S. Saito, S. Lombardi, S.-E. Wei, D. Belko, S.-I. Yu, Y. Sheikh, and J. Saragih. Authentic volumetric avatars from a phone scan. *ACM Trans. Graph.*, 41(4), July 2022.

[13] D. Z. Chen, Y. Siddiqui, H.-Y. Lee, S. Tulyakov, and M. Nießner. Text2tex: Text-driven texture synthesis via diffusion models. *arXiv preprint arXiv:2303.11396*, 2023.

[14] Y. Chen, L. Wang, Q. Li, H. Xiao, S. Zhang, H. Yao, and Y. Liu. Monogaussianavatar: Monocular gaussian point-based head avatar. *arXiv*, 2023.

[15] J. Chibane, D. Wolf, T. Hackel, and M. Nießner. Neural unsigned distance fields for implicit function learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[16] R. L. Cook and K. E. Torrance. A reflectance model for computer graphics. *ACM Trans. Graph.*, 1(1):7–24, Jan. 1982.

[17] P. Debevec, T. Hawkins, C. Tchou, H.-P. Duiker, W. Sarokin, and M. Sagar. Acquiring the reflectance field of a human face. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '00)*, pages 145–156. ACM, 2000.

[18] P. Debevec, T. Hawkins, C. Tchou, H.-P. Duiker, W. Sarokin, and M. Sagar. Acquiring the reflectance field of a human face. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, page 145–156, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[19] F. A. Fardo, V. H. Conforto, F. C. de Oliveira, and P. S. Rodrigues. A formal evaluation of psnr as quality measurement parameter for image segmentation algorithms, 2016.

[20] M. S. Floater and K. Hormann. Surface parameterization: a tutorial and survey. *Advances in Multiresolution for Geometric Modeling*, pages 157–188, 2005.

[21] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics: Principles and Practice*. Addison-Wesley Professional, 2nd edition, 1996.

[22] U. C. for Artificial Intelligence. Reconstructing environments using slam. `https://www.youtube.com/watch?v=oWWzCacRmYI`, 2023.

[23] S. Fridovich-Keil, A. Yu, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5501–5510, June 2022.

[24] 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)*, pages 8649–8658, June 2021.

[25] A. Ghosh, G. Fyffe, B. Tunwattanapong, J. Busch, X. Yu, and P. Debevec. Multiview face capture using polarized spherical gradient illumination. *ACM Trans. Graph.*, 30(6):1–10, Dec. 2011.

[26] S. Giebenhain, T. Kirschstein, M. Georgopoulos, M. Rünz, L. Agapito, and M. Nießner. Mononphm: Dynamic head reconstruction from monoculuar videos. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[27] S. Giebenhain, T. Kirschstein, M. Rünz, L. Agapito, and M. Nießner. Npga: Neural parametric gaussian avatars. In *SIGGRAPH Asia 2024 Conference Papers (SA Conference Papers '24), December 3-6, Tokyo, Japan*, 2024.

[28] P. Gotardo, J. Riviere, D. Bradley, A. Ghosh, and T. Beeler. Practical dynamic facial appearance modeling and acquisition. 37(6), Dec. 2018.

[29] K. Guo, P. Lincoln, P. Davidson, J. Busch, X. Yu, M. Whalen, G. Harvey, S. Orts-Escolano, R. Pandey, J. Dourgarian, D. Tang, A. Tkach, A. Kowdle, E. Cooper, M. Dou, S. Fanello, G. Fyffe, C. Rhemann, J. Taylor, P. Debevec, and S. Izadi. The relightables: volumetric performance capture of humans with realistic relighting. *ACM Trans. Graph.*, 38(6), Nov. 2019.

[30] A. Haque, M. Tancik, A. A. Efros, A. Holynski, and A. Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions, 2023.

[31] E. Hecht. *Optics*. Pearson Education, London, 5th edition, 2017.

[32] A. Horé and D. Ziou. Image quality metrics: Psnr vs. ssim. In *2010 20th International Conference on Pattern Recognition*, pages 2366–2369, 2010.

[33] B. Huang, Z. Yu, A. Chen, A. Geiger, and S. Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *SIGGRAPH 2024 Conference Papers*. Association for Computing Machinery, 2024.

[34] A. Jacobson, Z. Deng, K. Crane, E. Sifakis, L. Kavan, and F. Panozzo. libigl. `http://libigl.github.io`, Since 2012. A C++ geometry processing library.

[35] H. W. Jensen, S. R. Marschner, M. Levoy, and P. Hanrahan. A practical model for subsurface light transport. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '01)*, pages 511–518. ACM, 2001.

[36] J. T. Kajiya. The rendering equation. *SIGGRAPH Comput. Graph.*, 20(4):143–150, 1986.

[37] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila. Training generative adversarial networks with limited data. In *Proc. NeurIPS*, 2020.

[38] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, volume 256 of *SGP '06*, pages 61–70. Eurographics Association, 2006.

[39] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023.

[40] T. Kirschstein, S. Giebenhain, and M. Nießner. Diffusionavatars: Deferred diffusion for high-fidelity 3d head avatars. *arXiv preprint arXiv:2311.18635*, 2023.

[41] T. Kirschstein, S. Qian, S. Giebenhain, T. Walter, and M. Nießner. Nersemble: Multi-view radiance field reconstruction of human heads. *ACM Trans. Graph.*, 42(4), jul 2023.

[42] J. Li, C. Cao, G. Schwartz, R. Khirodkar, C. Richardt, T. Simon, Y. Sheikh, and S. Saito. Uravatar: Universal relightable gaussian codec avatars. In *ACM SIGGRAPH 2024 Conference Papers*, 2024.

[43] J. Li, D. Li, S. Savarese, and S. C. H. Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *ArXiv*, 2023.

[44] 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)*, 36(6):194:1–194:17, 2017.

[45] I. Loshchilov and F. Hutter. Decoupled weight decay regularization, 2019.

[46] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. G. Yong, J. Lee, W.-T. Chang, W. Hua, M. Georg, and M. Grundmann. Mediapipe: A framework for building perception pipelines. *ArXiv*, 2019.

[47] S. Ma, Y. Weng, T. Shao, and K. Zhou. 3d gaussian blendshapes for head avatar animation. In *ACM SIGGRAPH Conference Proceedings, Denver, CO, United States, July 28 - August 1, 2024*, 2024.

[48] P. Manu, A. Srivastava, and A. Sharma. Clip-head: Text-guided generation of textured neural parametric 3d head models. In *SIGGRAPH Asia 2023 Technical Communications*, SA '23, New York, NY, USA, 2023. Association for Computing Machinery.

[49] Maxon Computer GmbH. Cinema 4d. `https://www.maxon.net/`, Since 1990.

[50] M. Mendiratta, X. Pan, M. Elgharib, K. Teotia, M. B. R, A. Tewari, V. Golyanik, A. Kortylewski, and C. Theobalt. Avatarstudio: Text-driven editing of 3d dynamic human head avatars, 2023.

[51] 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 *ECCV*, 2020.

[52] T. Müller, A. Evans, C. Schied, and A. Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, July 2022.

[53] V. Müller. Polarization-based separation of diffuse and specular surface-reflection. In *DAGM-Symposium*, 1995.

[54] J. Munkberg, J. Hasselgren, T. Shen, J. Gao, W. Chen, A. Evans, T. Müller, and S. Fidler. Extracting Triangular 3D Models, Materials, and Lighting From Images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8280–8290, June 2022.

[55] F. E. Nicodemus. Directional reflectance and emissivity of an opaque surface. *Applied Optics*, 4(7):767, July 1965.

[56] A. Nicolai, M. Phan, A. Ruddle, and D. Smith. A facs valid 3d dynamic action unit database with applications to 3d dynamic morphable facial modeling. *International Conference on Computer Vision (ICCV)*, 2011.

[57] F. H. Pighin, J. Hecker, D. Lischinski, R. Szeliski, and D. Salesin. Synthesizing realistic facial expressions from photographs. *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, 1998.

[58] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv*, 2022.

[59] S. Qian. Versatile head alignment with adaptive appearance priors. September 2024.

[60] S. Qian, T. Kirschstein, L. Schoneveld, D. Davoli, S. Giebenhain, and M. Nießner. Gaussianavatars: Photorealistic head avatars with rigged 3d gaussians. *arXiv preprint arXiv:2312.02069*, 2023.

[61] S. Qian, T. Kirschstein, L. Schoneveld, D. Davoli, S. Giebenhain, and M. Nießner. Gaussianavatars: Photorealistic head avatars with rigged 3d gaussians. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20299–20309, 2024.

[62] S. Rahmann and N. Canterakis. Reconstruction of specular surfaces using polarization imaging. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages I–I, 2001.

[63] A. Ranjan, T. Bolkart, S. Sanyal, and M. J. Black. Generating 3d faces using convolutional mesh autoencoders. *CoRR*, abs/1807.10267, 2018.

[64] J. Riviere, P. Gotardo, D. Bradley, A. Ghosh, and T. Beeler. Single-shot high-quality facial geometry and skin appearance capture. *ACM Trans. Graph.*, 39(4), Aug. 2020.

[65] K. M. Robinette, C. L. Blackwell, H. A. M. Daanen, S. D. Fleming, C. C. Gordon, N. J. Paxton, B. D. Corner, and J. Roehlich. Caesar: Summary statistics for the us, italian, and dutch databases. *SAE Technical Paper*, 1999.

[66] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, June 2022.

[67] S. Saito, G. Schwartz, T. Simon, J. Li, and G. Nam. Relightable gaussian codec avatars. In *CVPR*, 2024.

[68] P. V. Sander, J. Snyder, S. J. Gortler, and H. Hoppe. Texture mapping progressive meshes. *ACM Transactions on Graphics (TOG)*, 20(3):163–192, July 2001.

[69] R. Sawhney and K. Crane. Boundary first flattening. *ACM Transactions on Graphics (TOG)*, 36(4), July 2017. Presented at SIGGRAPH 2017.

[70] C. Schlick. An inexpensive brdf model for physically-based rendering. *Computer Graphics Forum*, 13(3):233–246, 1994.

[71] C. Schuhmann, R. Beaumont, R. Vencu, C. Gordon, R. Beaumont, L. McLaughlin, A. Blattmann, K. Seonghyeon, M. Romero, R. Karumbaiah, K. Ligett, M. Wortsman, B. Ommer, and L. Schmidt. Laion-5b: An archive of millions of open internet pictures with clip embeddings. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 2527–2540, 2022.

[72] Z. Shao, Z. Wang, Z. Li, D. Wang, X. Lin, Y. Zhang, M. Fan, and Z. Wang. SplattingAvatar: Realistic Real-Time Human Avatars with Mesh-Embedded Gaussian Splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[73] A. Sheffer, E. Praun, and K. Rose. Mesh parameterization methods and their applications. *Foundations and Trends in Computer Graphics and Vision*, 2(2):105–171, 2006.

[74] A. D. M. N. Shivangi Aneja, Justus Thies. Clipface: Text-guided editing of textured 3d morphable models. In *ACM SIGGRAPH 2023 Conference Proceedings*, SIGGRAPH '23, 2023.

[75] M. G. M. R. L. A. M. N. Simon Giebenhain, Tobias Kirschstein. Learning neural parametric head models. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[76] B. Smith. Geometrical shadowing of a random rough surface. *IEEE Transactions on Antennas and Propagation*, 15(5):668–671, 1967.

[77] SNS Insider. Augmented and virtual reality market set to reach usd 237.0 billion by 2032. *GlobeNewswire (Publishing the report summary)*, November 2024.

[78] Y. Song, S. Garg, J. Wu, Y. Xu, J. Zhang, and S. Ermon. Score-based generative modeling with stochastic differential equations. *International Conference on Learning Representations (ICLR)*, 2021. arXiv:2011.13456.

[79] A. Srivastava, P. Manu, A. Raj, V. Jampani, and A. Sharma. Wordrobe: Text-guided generation of textured 3d garments, 2024.

[80] M. Srámek and F. Beleznay. 3d distance fields: A survey of techniques and applications. *IEEE Transactions on Visualization and Computer Graphics*, 12(4):559–572, 2006.

[81] L. Tran and X. Liu. Nonlinear 3d face morphable model. *CoRR*, abs/1804.03786, 2018.

[82] D. Vlasic, M. Brand, H. Pfister, and J. Popović. Multilinear models for face synthesis. 01 2004.

[83] B. Walter, S. R. Marschner, H. Li, and K. E. Torrance. Microfacet models for refraction through rough surfaces. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques*, EGSR'07, page 195–206, Goslar, DEU, 2007. Eurographics Association.

[84] C. Wang, D. Kang, H.-Y. Sun, S.-H. Qian, Z.-X. Wang, L. Bao, and S.-H. Zhang. Mega: Hybrid mesh-gaussian head avatar for high-fidelity rendering and head editing. *arXiv preprint arXiv:2404.19026*, 2024.

[85] Y. Wang, C. Wang, B. Gong, and T. Xue. Bilateral guided radiance field processing, 2024.

[86] C. A. Wilson, A. Ghosh, P. Peers, J.-Y. Chiang, J. Busch, and P. Debevec. Temporal upsampling of performance geometry using photometric alignment. *ACM Trans. Graph.*, 29(2), Apr. 2010.

[87] L. Wolff and T. Boult. Constraining object features using a polarization reflectance model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(7):635–657, 1991.

[88] R. J. Woodham. Photometric method for determining surface orientation from multiple images. 1980.

[89] M. Wu, H. Zhu, L. Huang, Y. Zhuang, Y. Lu, and X. Cao. High-fidelity 3d face generation from natural language descriptions. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[90] J. Xiang, X. Gao, Y. Guo, and J. Zhang. Flashavatar: High-fidelity head avatar with efficient gaussian embedding, 2024.

[91] Y. Xu, B. Chen, Z. Li, H. Zhang, L. Wang, Z. Zheng, and Y. Liu. Gaussian head avatar: Ultra high-fidelity head avatar via dynamic gaussians. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[92] Y. Xu, J. Riviere, G. Zoss, P. Chandran, D. Bradley, and P. Gotardo. Improved Lighting Models for Facial Appearance Capture. In N. Pelechano and D. Vanderhaeghe, editors, *Eurographics 2022 - Short Papers*. The Eurographics Association, 2022.

[93] L. Zhang and M. Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3836–3847, October 2023.

[94] X. Zhao, L. Wang, J. Sun, H. Zhang, J. Suo, and Y. Liu. Havatar: High-fidelity head avatar via facial model conditioned neural radiance field. *ACM Trans. Graph.*, oct 2023. Just Accepted.

[95] Y. Zheng, W. Yifan, G. Wetzstein, M. J. Black, and O. Hilliges. Pointavatar: Deformable point-based head avatars from videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.