VIETNAM NATIONAL UNIVERSITY, HANOI
UNIVERSITY OF ENGINEERING AND TECHNOLOGY

Đào Quang Hiếu

MÔ HÌNH SINH AVATAR 3D BAO GỒM TÓC VÀ MẶT TỪ MỘT ẢNH DUY NHẤT

A SINGLE-IMAGE 3D AVATAR RECONSTRUCTION MODEL INCLUDING HAIR AND FACE

Major: Information Technology (Honors Program)

HA NOI - 2023

VIETNAM NATIONAL UNIVERSITY, HANOI
UNIVERSITY OF ENGINEERING AND TECHNOLOGY

Đào Quang Hiếu

MÔ HÌNH SINH AVATAR 3D BAO GỒM TÓC VÀ MẶT TỪ MỘT ẢNH DUY NHẤT

A SINGLE-IMAGE 3D AVATAR RECONSTRUCTION MODEL INCLUDING HAIR AND FACE

Major: Information Technology (Honors Program)

Supervisor: Dr. Ma Thị Châu

HA NOI - 2023

AUTHORSHIP

"I hereby declare that the work contained in this thesis is of my own and has not been previously submitted for a degree or diploma at this or any other higher education institution. To the best of my knowledge and belief, the thesis contains no materials previously published or written by another person except where due reference or acknowledgment is made."

Signature

Đào Quang Hiếu

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to Dr. Ma Thi Chau for her invaluable guidance and support in shaping my thesis. Her expertise and dedication have been instrumental in developing my research idea, and I am deeply thankful for her mentorship.

ABSTRACT

The development of computer graphics and machine learning has propelled remarkable advancements in the creation of high-precision and aesthetically pleasing 3D avatars. From the early days of computing in the 1950s to the present day, we have witnessed the potent synergy between computer graphics and machine learning in generating 3D products that find application across numerous domains. In this thesis, I will present a system that can generate 3D avatars from a single-view image. This system uses a combination of multiple state-of-the-art methods and my proposed network architecture and pipeline to reconstruct a highly accurate and customizable 3D avatar of a person's head. Surveys have been conducted and show positive results TODO: fix ambiguousness.

Keywords: Computer Vision, Neural Network, 3D Face Reconstruction, 3D Morphable Model

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

DIFD Domain Invariant Feature Descriptors

GAN Generative Adversarial Network

HMI Human-machine Interaction

LBS Linear Blend Skinning

LPIPS Learned Perceptual Image Patch Similarity

MSE Mean Squared Error

NeRF Neural Radiance Fields

PCA Principal Component Analysis

PSNR Peak Signal-to-Noise Ratio

SSIM Structural Similarity Index Measure

VAE Variational Autoencoder

VR Virtual Reality

Chapter 1

INTRODUCTION

1.1 Motivation

The ability to create 3D representations of oneself, namely, 3D avatars, has gained the attention of the crowd lately. From the non-research groups that have needs for their self-avatar creation to the researchers who actively work in related fields, it appears that the attention on that is much higher than that of a decade ago. A simple explanation for that is that 3D avatar technology has found its way into practical usage.

First, with the emergence of Virtual Reality (VR) technology, people now want to see others in the virtual worlds more vividly than in non-VR 3D scenarios. That means they want their and others' avatars to express emotions freely, and to be able to represent their personas accurately. Secondly, traditional methods of creating a 3D scene in animation involve manually constructing 3D characters with 3D creation software. That usually costs a lot of money and time, as 3D graphic work requires skills and hundreds of hours to create satisfactory 3D objects. The traditional methods often give better output, but for some people that can be overkill. Moreover, using a lot of money to hire people to create 3D works can be detrimental to certain companies' financial situation. These two reasons can be why the automated approaches to 3D avatar reconstruction/creation are emerging.

Therefore, I've been researching methods that can simplify or automatically reconstruct 3D avatars from limited input. In the process of researching the best solution to this problem, I found that machine-learning methods can output great results for generative works. With the support of Dr. Ma Thi Chau and the Human-machine Interaction (HMI) laboratory, I was able to create a system for automated 3D avatar reconstruction and improve it gradually using machine learning methods. The system was then evaluated and brought into use, and achieved great results (which will be elaborated in Chapter 4).

Thanks to all the support I've received, especially from Dr. Chau, I was able to present this system in ICTA 2023 - an international conference on Advances in Information and Communication Technology.

1.2 Contributions and thesis overview

1.2.1 Contributions

The contributions of the thesis involve the creation of the proposed system, which are:

- A novel pipeline for handling the 3D reconstruction of avatars from a single-view image, where the hair is created uniquely, separated from the head model.
- A method to transfer basic, straightforward human emotions to FLAME - a 3D morphable model's - parameters, or easily customizable emotions.

1.2.2 Thesis overview

The rest of this thesis is organized as follows:

Chapter 2 provides the related work and fundamentals that are applied to the pipeline of the proposed system.

In chapter 3, each step of the proposed system's pipeline is explained in detail and with mathematical formulas.

Chapter 4 provides quantitative results of the working system from surveys of the system's users and the experts, and qualitative results in common and specialized metrics.

Chapter 5 concludes the thesis and provides future work.

Chapter 2

RELATED WORK

2.1 3D avatar reconstruction

2.1.1 3D face reconstruction

Creating a 3D model of a human head can be done using various methods, ranging from manual to fully automated. Manual methods involve using 3D modeling software such as Blender, Autodesk Maya, or ZBrush to create a model from scratch, using techniques such as sculpting or modeling with geometric primitives. Less manual methods involve starting with a base head model and making changes to it.

The concept of 3D Morphable Models (3DMMs) was introduced by Blanz and Vetter [1], which represented the shape and texture variations of faces using linear statistical models, specifically Principal Component Analysis (PCA). This method allows for the formalization of the diversity of human faces using a small number of parameters. Various works [2]–[6] have been dedicated to creating a generalized 3DMM.

To better express facial details, recent works have introduced non-linearity by integrating neural networks into 3DMMs such as VAE [7], GAN [8], or NeRF [9], [10].

2.1.1.1 FLAME

As the high-end methods for generating 3D faces require extensive labor and the low-end methods lack facial expressiveness, FLAME [5] aims to be a middle ground for 3D face modeling. FLAME is a 3DMM model that can reproduce realistic and expressive 3D face models that accurately capture the variations in facial shape and expression. FLAME separates the representation of identity, pose, and facial expression into different parameter spaces and combines them with linear blend skinning (LBS) and blendshapes. Its ability to reproduce 3D face models with high expressiveness has made it the foundation for many state-of-the-art face reconstruction models.

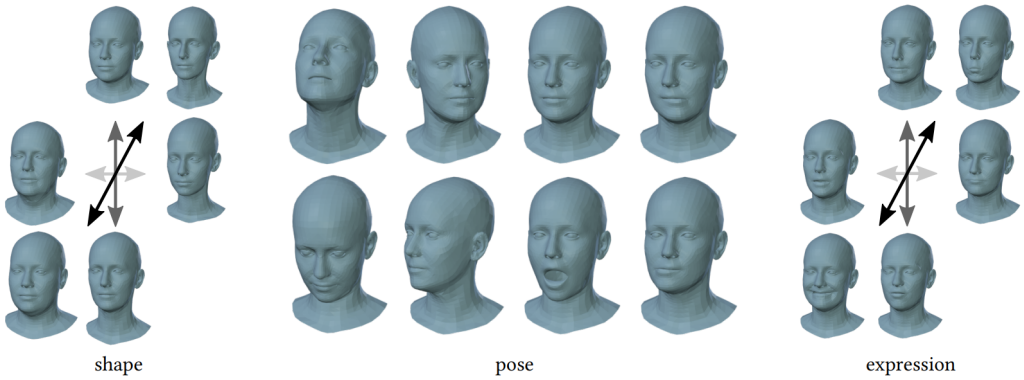

Figure 2.1: Different types of FLAME parameters for controlling the 3D shape

2.1.1.2 DECA

DECA [11] is a method to reconstruct 3D face models from a single-view image, using FLAME as a component in the process of reconstructing the 3D model. In addition to detecting the facial shape and expression, DECA can map the facial texture from the image to the 3D model using a 3D texture space.

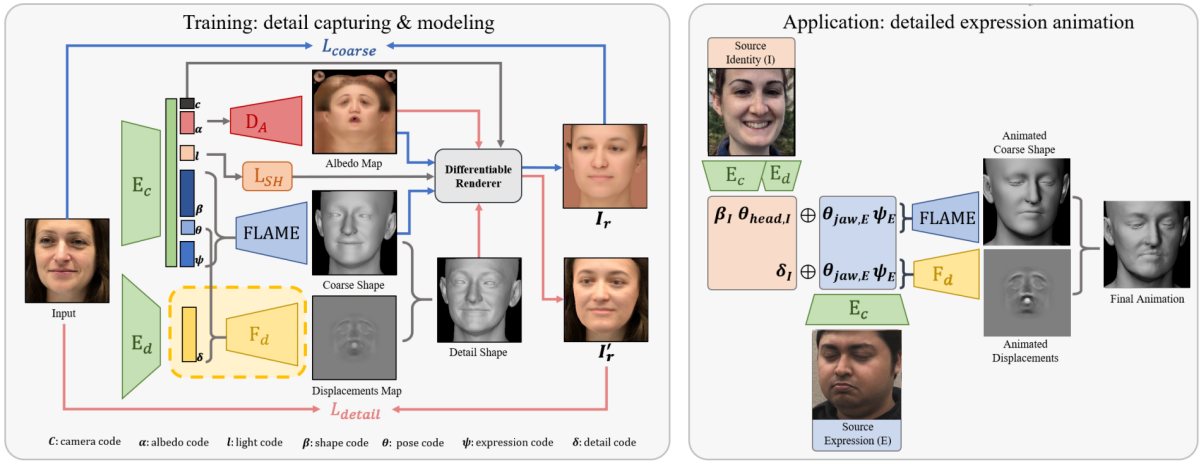

Figure 2.2: DECA architecture, using FLAME as part of the pipeline

Given an input image, DECA encoders output

- FLAME parameters including: shape parameters (β), expression parameters (ψ), pose parameters (θ)
- Detail parameters (δ)

- Albedo parameters (α)
- Camera parameters (C)
- Light parameters (l)

The FLAME parameters are used to construct the FLAME 3D mesh, in which the shape parameters define the invariant mesh identity, whereas the expression parameters and pose parameters define the variant mesh expression and pose. The detail parameters are used to reconstruct the high-frequency details of the face. The albedo parameters are used to reconstruct the face texture. The camera parameters are used to reconstruct the camera pose. The light parameters are used to reconstruct the lighting conditions.

The 3D face model is then rendered to form the 2D face image. The 2D face image is then compared with the input image to calculate the loss. The loss is then used to update the parameters. The parameters are then used to reconstruct the 3D face model. The process is repeated until the loss is minimized.

2.1.1.3 MICA

MICA [12] is a method to reconstruct 3D face models from a single-view image, which also uses FLAME as a component in the process of reconstructing the 3D model. However, unlike DECA, MICA only reconstructs the 3D face model with the shape parameters (β) and does not reconstruct the facial expression as well as the texture.

MICA uses a different approach to reconstruct the shape parameters compared to DECA. MICA bases its network on the ArcFace architecture [13] and refines the last 3 ResNet blocks of the architecture. This results in a more accurate reconstruction of the shape parameters compared to DECA, as shown in the NoW benchmark [14].

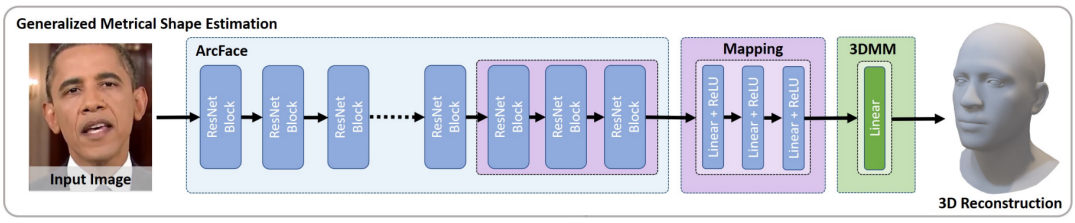

Figure 2.3: MICA architecture.

2.1.2 3D hair reconstruction

Representing hair structure in a three-dimensional environment is a complex task [15]. Several studies, such as [16]–[19] represent hair as a mesh. While this representation serves specific

purposes, it still poses various limitations such as refinement, animation, rendering, and so on. Other techniques have been developed for higher-quality 3D hair modeling [15]. These include clustering hair into fiber groups and representing it as cylinders [20] or modeling each hair strand individually [15]. Modeling each hair strand fulfills requirements for practical applications. The latest research focused on hair modeling from images [21]. This includes techniques for creating a 3D hair model from multiple images [22], as well as from a single image [23]–[25].

2.1.2.1 HairNet

HairNet [26] was the pioneering deep learning-based model for reconstructing 3D hair from a single image. HairNet employed data augmentation techniques to create a large dataset comprising 40, 000 hairstyles. Its model architecture followed an encode-decode model, where the input was encoded into a feature vector and then decoded back into a 3D hair model. HairNet's innovative use of synthetic data for training purposes has been adopted by subsequent models. HairNet applied a 2D capture for each synthetic hairstyle and transformed it into an intermediate format called an oriented map. The oriented map provides directional information for the model.

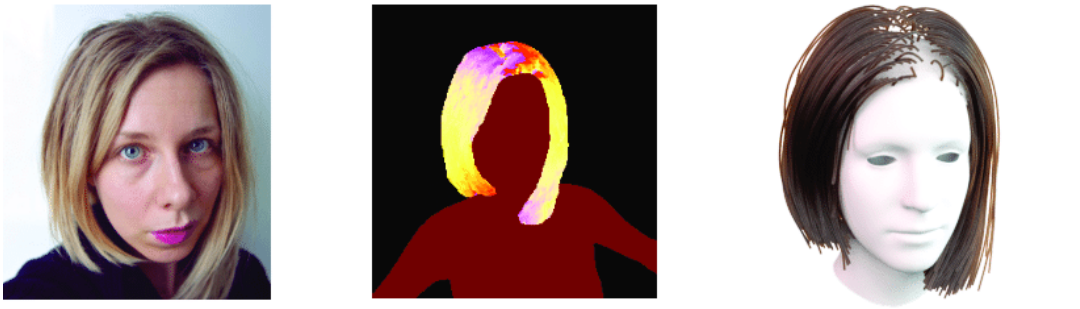

Figure 2.4: HairNet orientation map detection and reconstruction output.

2.2 Emotion customization

2.2.1 Overview

Using the FLAME model, the input parameters are categorized into 3 groups: shape parameters, expression parameters, and pose parameters. To change the facial expression, one would apply changes to FLAME expression parameters and pose parameters. However, these expression parameters are non-descriptive and are too many which can make the users confused. Therefore, more simple and descriptive parameters are needed for representing basic human emotions.

Based on the common need for customizing facial emotion, a set of 6 basic emotions $S_e = \{happiness, anger, sadness, fear, contempt, surprise\}$ is defined to customize the facial expression. These emotions are defined in the Arousal-Valence Model and are common for basic usage.

The parameters responsible for dictating the facial expression in FLAME are expression parameters and pose parameters. To map these emotions to FLAME parameters, given that FLAME parameters mostly use linear morphing, one idea is to use a basic multi-layer perceptron architecture. The model implementing this should take the intensity of these emotions in the range of $[0, 1]$ and return the corresponding FLAME parameters that are used for emotions.

Chapter 3

THE METHOD

3.1 Requirements analysis

3.1.1 Introduction

This thesis aims to create a system that can reconstruct a 3D avatar from a single-view portrait image. The system should be able to handle a variety of tasks related to 3D avatar creation, including:

- Creating a 3D avatar from a single-view portrait image
- Customizing the 3D avatar's facial expression
- Trying on different hairstyles on the 3D avatar

From the requirements above, an analysis of the system's requirements is conducted to determine the system's architecture and the methods used to create the system. This section clarifies the requirements analysis and the system's architecture, using UML diagrams.

3.1.2 Use cases

The use cases of the system are shown in the figure below:

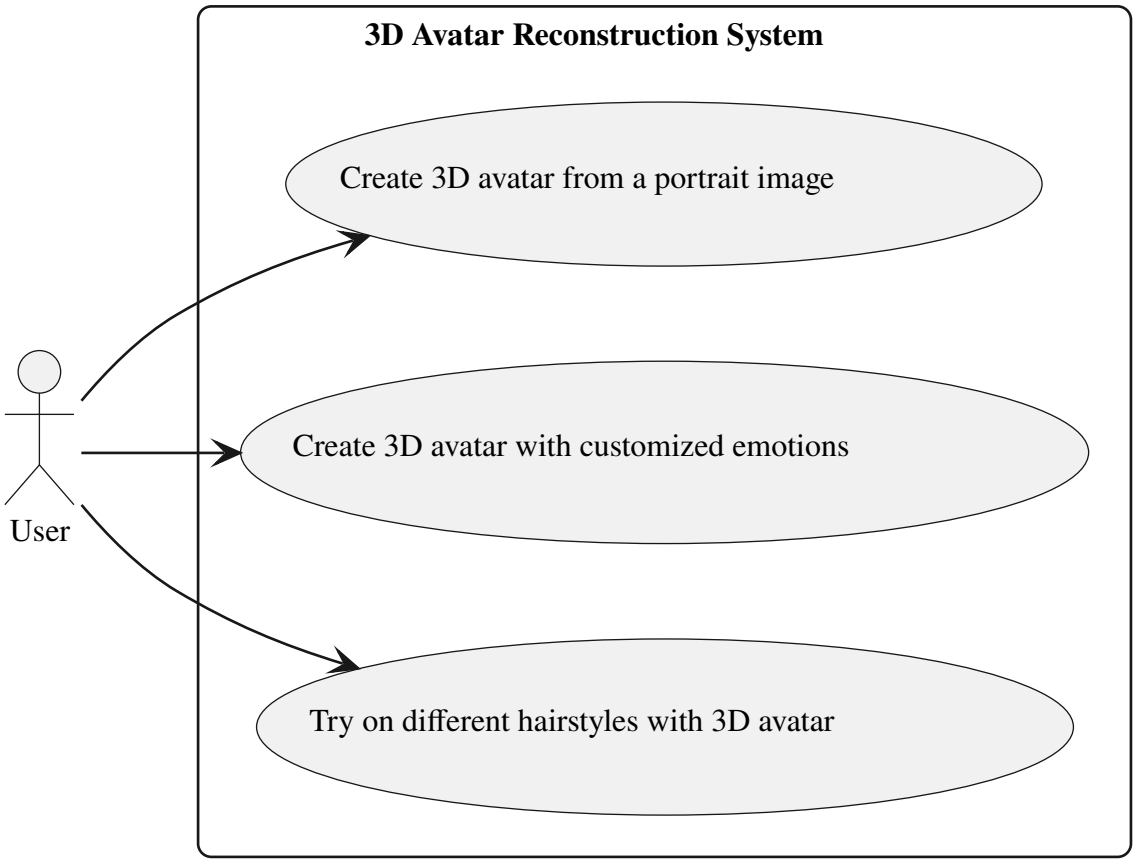

Figure 3.1: Use cases of the system

3.1.2.1 Create 3D avatar from a portrait image

This use case is the main use case of the system. The system should provide a GUI (graphical user interface) to allow the user to upload their images easily. The user uploads a single-view portrait image to the system, and the system will process the image and output a 3D avatar in the form of a 3D mesh reconstructed from the input image. The user can choose to download the 3D avatar as a zip file, which contains the 3D avatar model and texture map. The sequence diagram of this use case is shown in the figure below.

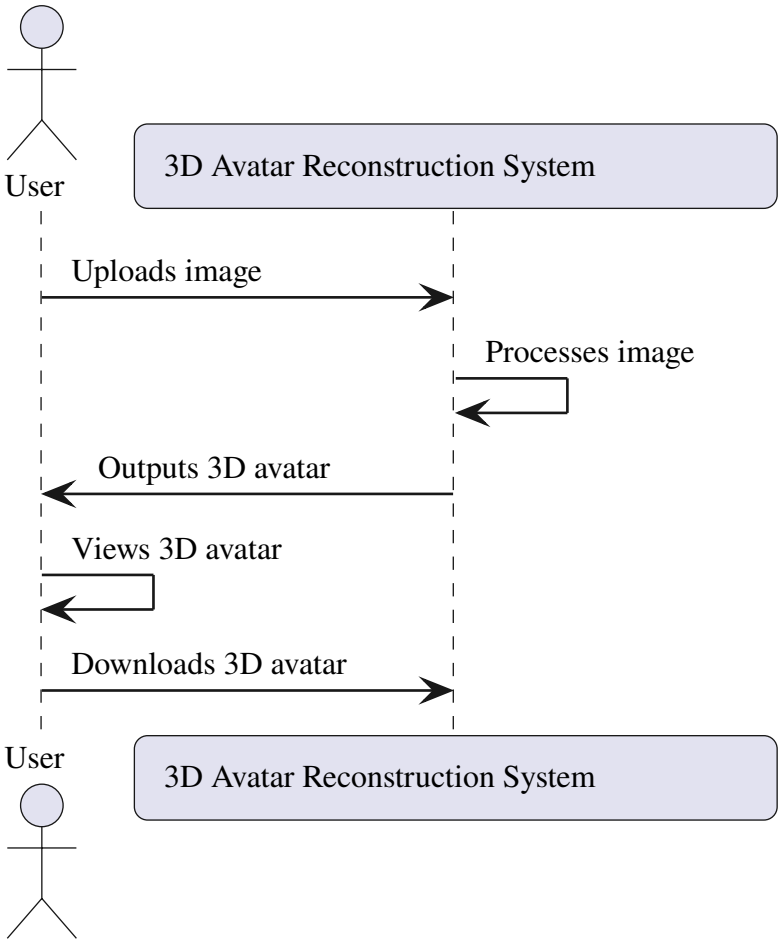

Figure 3.2: Analysis: Sequence diagram: Create 3D avatar from a portrait image.

3.1.2.2 Create 3D avatar with customized emotions

This use case is an extension of the main use case. The user can choose to customize the 3D avatar's emotion by using sliders to adjust the intensity of a set of emotions. The system will then output a 3D avatar with the emotions applied. The user can choose to download the 3D avatar as a zip file, which contains the 3D avatar model and texture map. The sequence diagram of this use case is shown in the figure below.

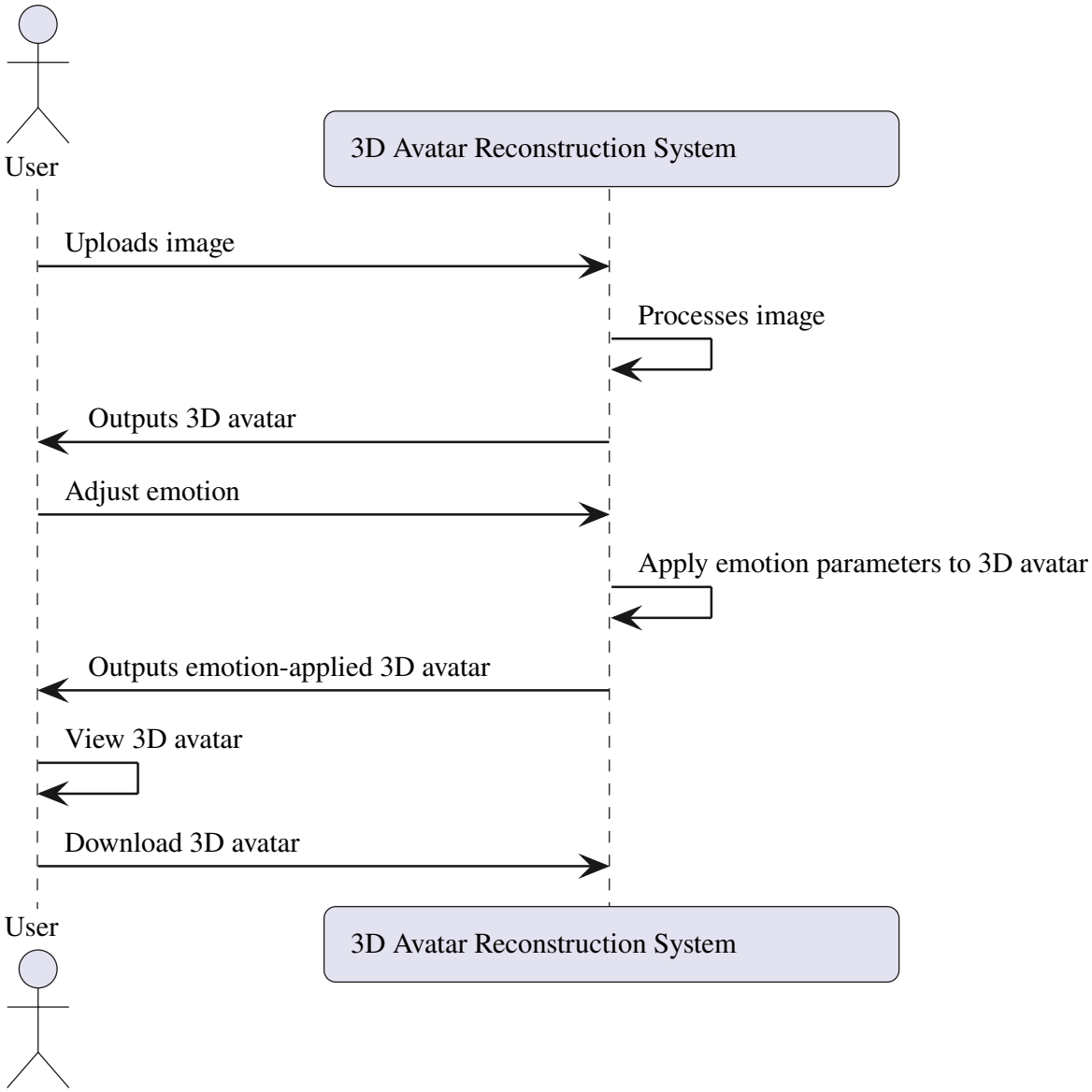

Figure 3.3: Analysis: Sequence diagram: Create 3D avatar with customized emotions.

3.1.2.3 Try on different hairstyles with 3D avatar

This use case is an extension of the main use case. The user can choose to try on different hairstyles with the 3D avatar. The system will then output a 3D avatar with the selected hairstyle. The user can choose to download the 3D avatar as a zip file, which contains the 3D avatar model, the texture map, and the hairstyle model. The sequence diagram of this use case is shown in the figure below.

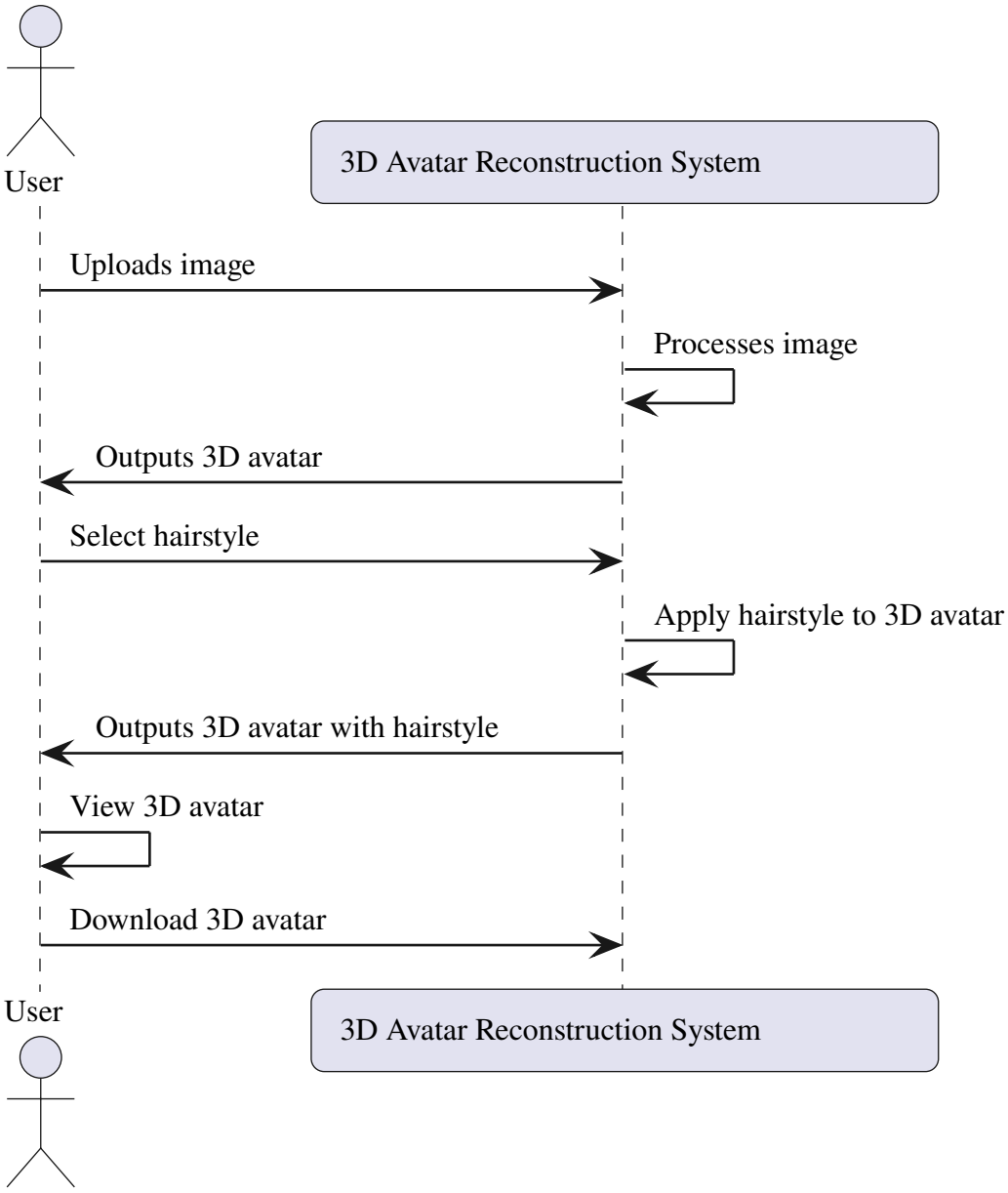

Figure 3.4: Analysis: Sequence diagram: Try on different hairstyles with 3D avatar.

3.2 System architecture

3.2.1 Overall flow

Essentially, the system architecture serves the purpose of taking a single-view portrait image of a person and outputs a 3D avatar reconstructed from the input image. The overview of the system flow and the decision tree corresponding to the user's options are shown in the figure below.

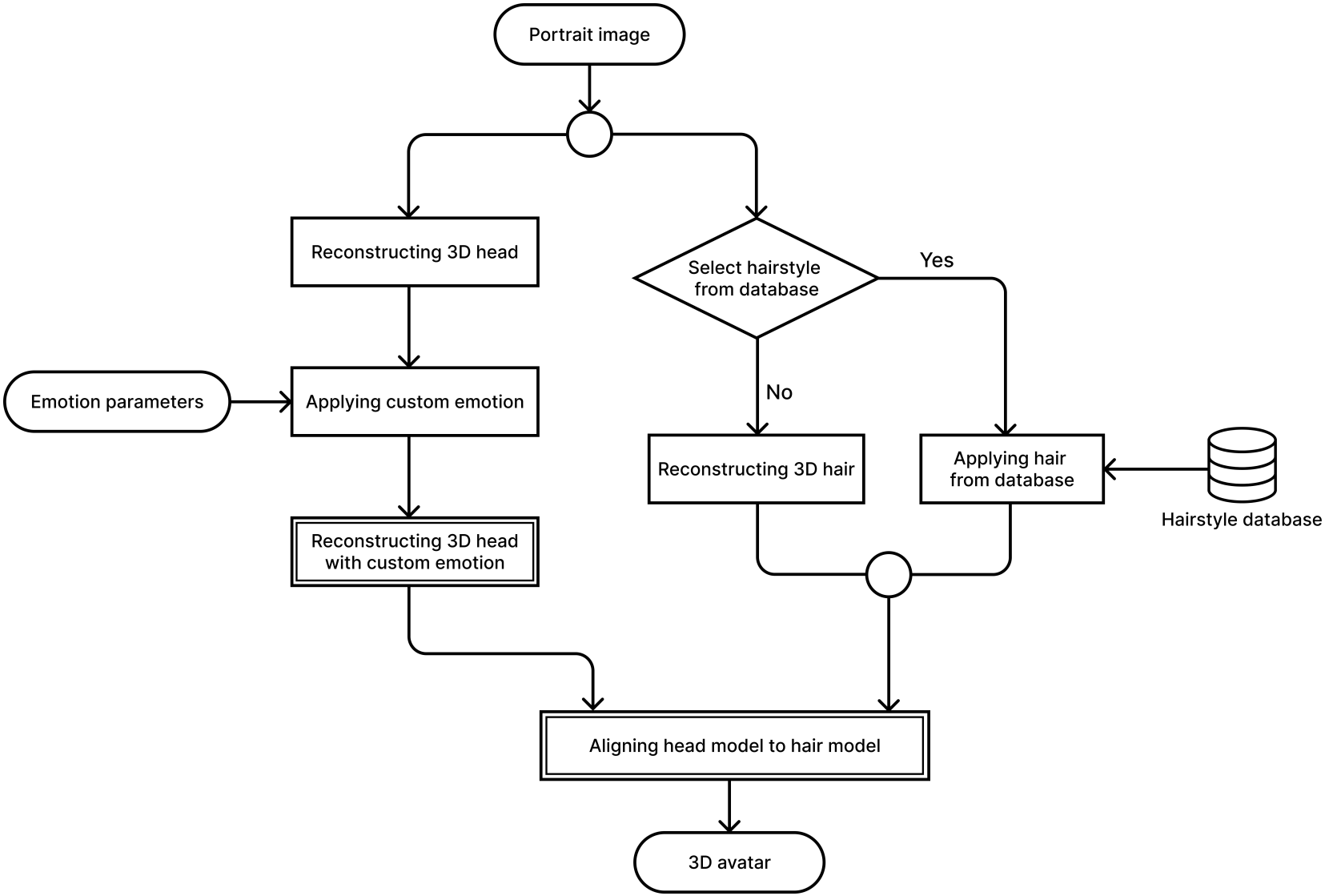

Figure 3.5: The system flow overview, with options for reconstruction output.

The system holds a database for hairstyles, which is convenient for try-on purposes. Instead of going through the standard flow where the system extracts the user's hairstyle from the captured image, the user can try on a variety of hairstyles in the database to see if any of these hairstyles suit their face.

The details of each reconstruction block will be explained in detail in the sections below, where the double-bordered blocks are the blocks that are novel and implemented in this thesis.

3.2.2 Flow for use case: Create 3D avatar from a portrait image

For taking a single-view portrait image as input, the system provides a front-end interface, i.e. a web page, where the user can upload their image. The front-end interface allows the user to send the image to a gateway server, which is responsible for handling the user's requests and sending the image to the back-end server for processing. By using a gateway server, the system architecture can be modularized, which means multiple reconstruction backends can be used as substitutions. A gateway also allows the system to easily scale up to handle a large number of requests by adding more back-end servers.

The back-end server is responsible for reconstruction tasks and algorithmic tasks. To be able to create a 3D avatar from a single-view portrait image, the system needs to be able to reconstruct the 3D face model from the input image. The system uses a pre-trained DECA model to reconstruct the 3D face shape, expression, and texture. DECA can be combined or substituted by the pre-trained MICA model [12] to reconstruct the 3D face shape using an alternative method. The algorithmic details of DECA and MICA will be explained in the next section 3.2.2.1.

The implemented flow for creating a 3D avatar from a single-view portrait image is shown in the figure below.

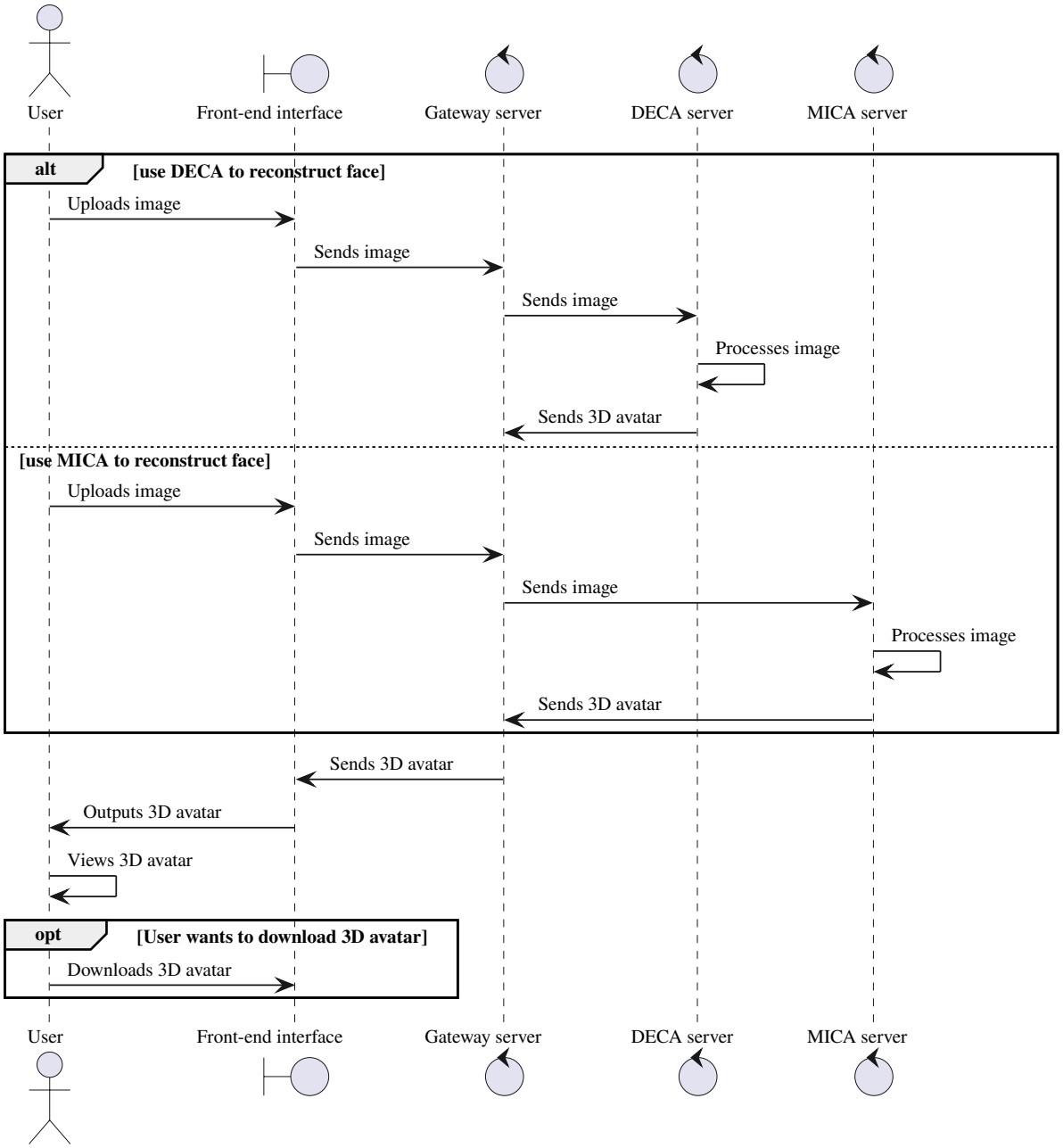

Figure 3.6: Implementation: Sequence diagram: Create 3D avatar from a portrait image.

3.2.2.1 Face reconstruction

To generate the 3D head, we used the combination of DECA [11] and MICA [12]. DECA takes a single input image and estimates the parameters for the FLAME model, which are identity's shape (β), head pose (θ), and expression (ψ), which then outputs a mesh of 5023 vertices. At the time of creating DECA, FLAME didn't have an albedo model for the reconstruction of skin texture, therefore DECA used Basel Face Model's albedo space for texture reconstruction. Since then, there have been works that make it possible to construct FLAME texture space [27], [28]. While it is possible to use DECA for the identity's shape (β) parameters, we decided to use MICA for the fitting of the shape parameters as MICA can reconstruct these parameters with metrical accuracy and produces generally better results, as demonstrated in the NoW Challenge [14] benchmarks.

3.2.2.2 Face and hair alignment

To make our system a fully automated avatar reconstruction system, the hair model and face model need to be automatically aligned with each other. To achieve this, we propose an automated procedure for aligning the hair model with the face model. First, we define the anchor point of the hair object and the head object (Fig 3.7) as the highest point of the mesh and the center of the mesh, respectively, from the top view.

For the head mesh $M_{head} = \{p_1,\ p_2, ...,\ p_N\}$, we sample 5% of the points with the highest z-axis values to create the set $M_{head\_5\%}$. For the hair mesh M_{hair}, for every hair strand, we sample 5% of the points that start from the strand root to create the set $M_{hair\_5\%}$. We find that 5% is enough to sample the points that form the head top, which can be used to approximate the center of the head and the hair accurately.

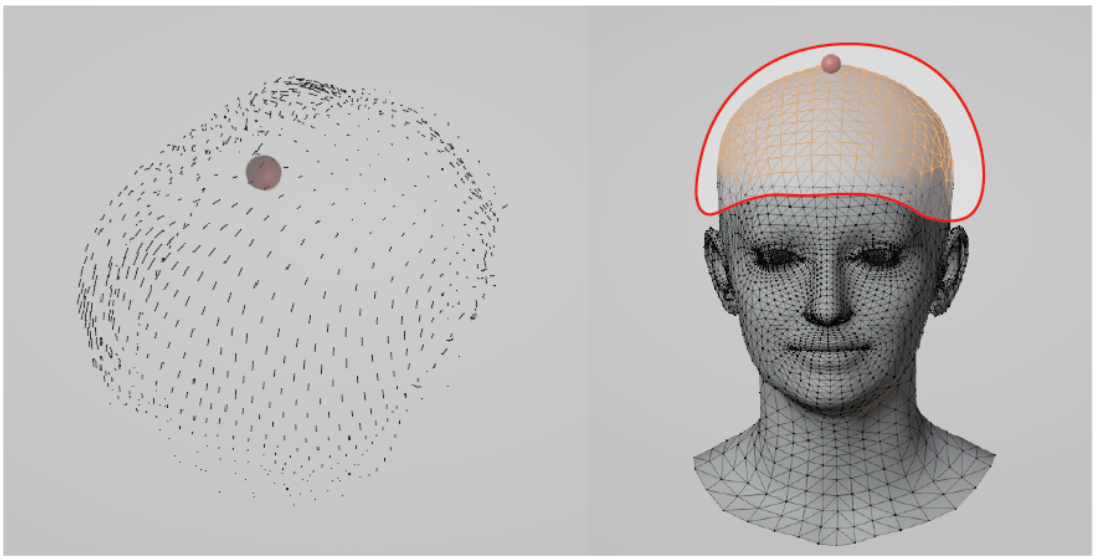

Figure 3.7: Sampled points in hair and head models to calculate the anchor points.

We then calculate the hair anchor point A_{hair} and the head anchor point A_{head} by taking the means of the x-axis and the y-axis and the highest value of the z-axis.

$$A = \begin{pmatrix} x_{mean\_5\%} \\ y_{mean\_5\%} \\ z_{max\_5\%} \end{pmatrix} = \begin{pmatrix} \frac{x_{p1}+x_{p2}+...+x_{pN5\%}}{0.05N} \\ \frac{y_{p1}+y_{p2}+...+y_{pN5\%}}{0.05N} \\ \max(z_{p1}, z_{p2}, ..., y_{pN_{5\%}}) \end{pmatrix} \tag{1}$$

With the hair anchor and the head anchor, the translation vector T_{hair} (Fig 3.8) is calculated by taking the subtraction of the 2 anchor points and adding it by a constant calibration value C.

$$T_{hair} = A_{hair} - A_{head} + C \tag{2}$$

For the scaling adjustment, first, I find the axis-aligned bounding boxes of the hair model and the head model. An assumption is made, which is the head model and the hair model are front-facing, as per the requirement of the captured image. Thus, the front-view lengths of the hair and the head can be deduced by taking the x-axis lengths. With the hair and the head length, the scaling coefficient can be calculated by taking the ratio of the hair length to the head length and can be adjusted with the addition constant C.

$$S_{hair} = \frac{W_{hair}}{W_{head}} + C \tag{3}$$

The final adjustment formula can be calculated with the translation vector T_{hair} and the scaling coefficient S_{hair}

$$p_{new} = \frac{1}{S_{hair}} \cdot (p_{old} + T_{hair}) \tag{4}$$

By aligning the hair model with the already aligned head model, the problem of HairNet being unable to align with the captured image as mentioned in Xu et al.'s paper [29] is addressed.

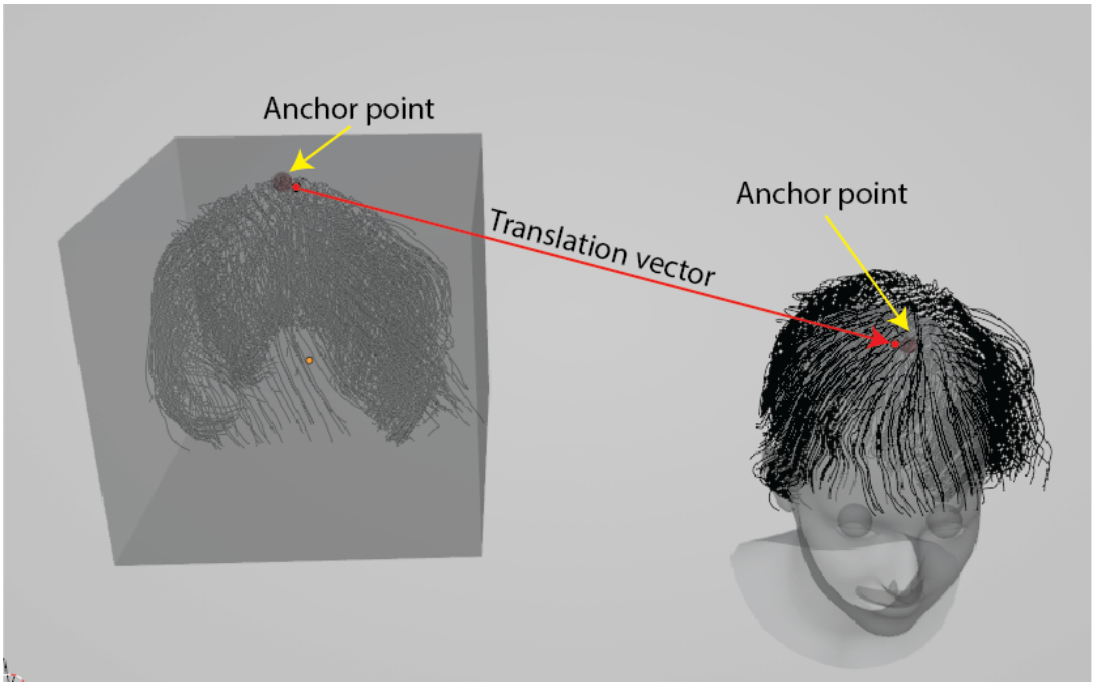

Figure 3.8: The alignment of the hair model with the head model by translation.

3.2.3 Flow for use case: Create 3D avatar with customized emotions

The system allows the user to customize the 3D avatar's emotion by using sliders to adjust the intensity of a set of emotions. The system uses a simple emotion-to-FLAME regressive model to convert the emotion parameters to FLAME's pose and expression parameters. The emotion-to-FLAME regressive model is trained on the VKIST dataset, with the ground truth acquired by running the dataset through the pre-trained face reconstruction model (DECA, MICA). The implemented flow for creating a 3D avatar with customized emotions is shown in the figure below.

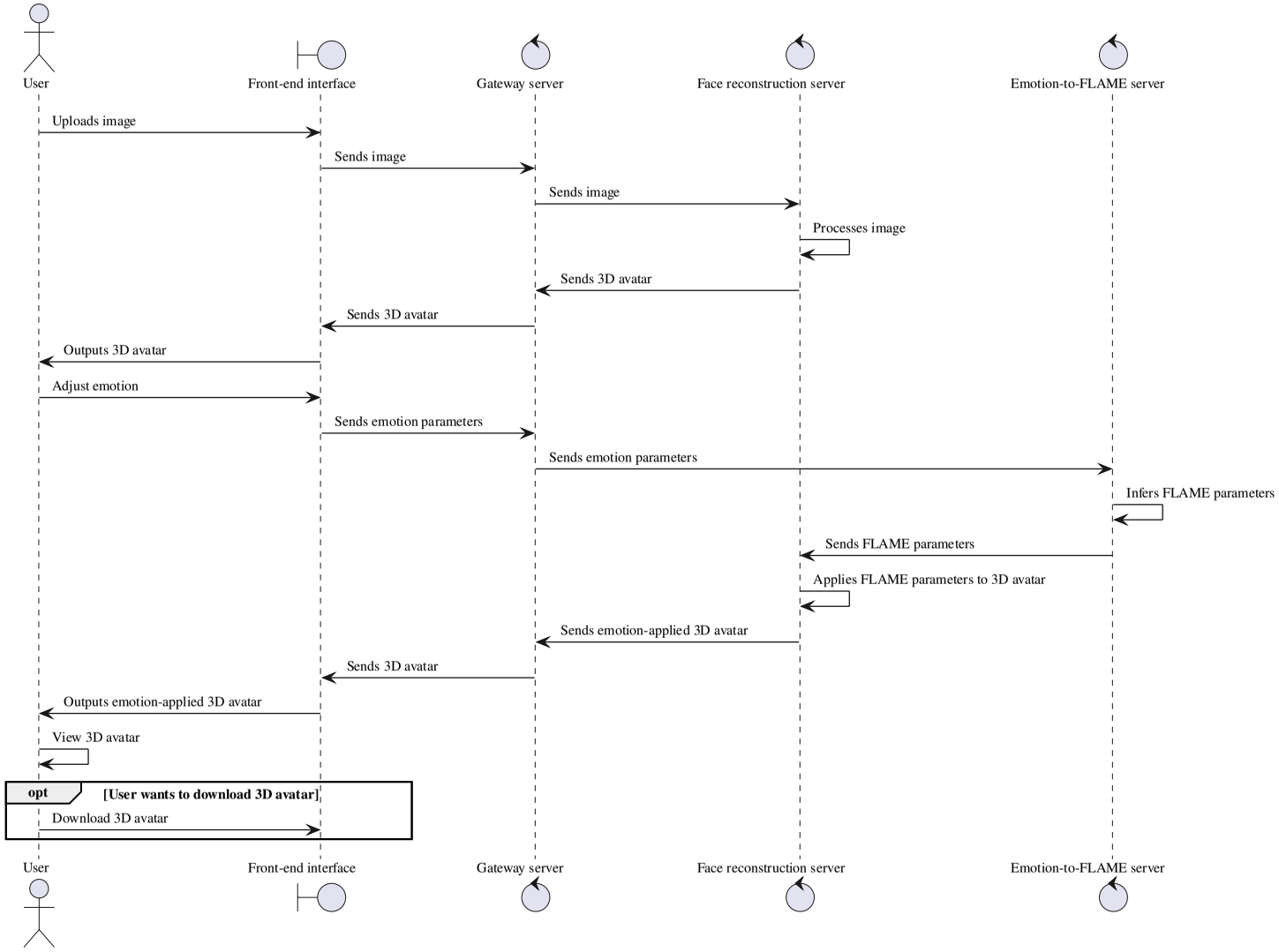

Figure 3.9: Implementation: Sequence diagram: Create 3D avatar with customized emotions.

3.2.4 Flow for use case: Try on different hairstyles with 3D avatar

The system allows the user to try on different hairstyles with the 3D avatar. The system uses a pre-trained image-to-hair model to reconstruct the user's hairstyle from the input image. If the user chooses to try on different hairstyles, the system will not use the user's hairstyle from the input image. Instead, the system will use one of the hairstyles in the database. The user can choose to download the 3D avatar with the hairstyle as a zip file, which contains the 3D avatar model, the texture map, and the hairstyle model. The implemented flow for trying on different hairstyles with the 3D avatar is shown in the figure below.

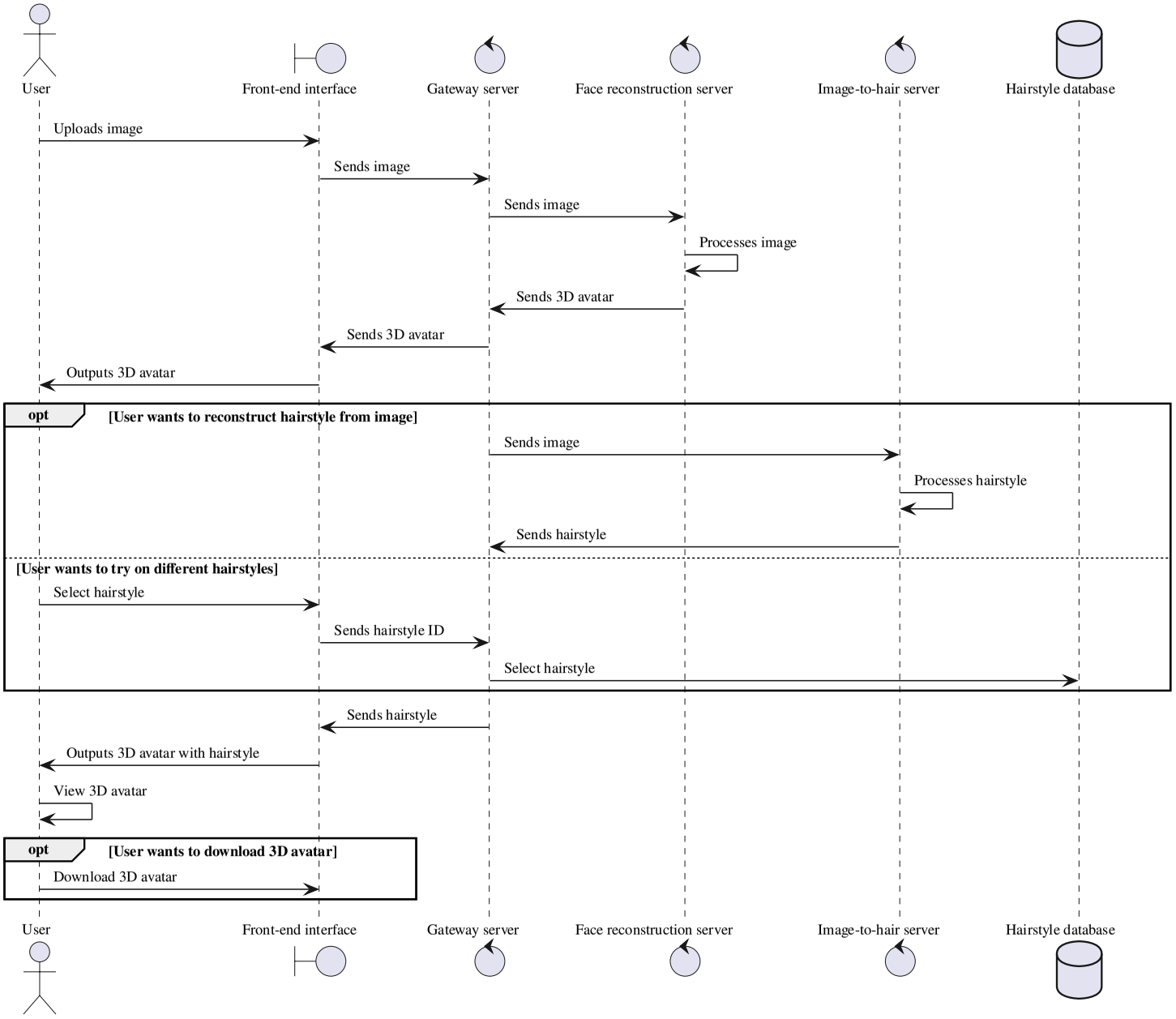

Figure 3.10: Implementation: Sequence diagram: Try on different hairstyles with 3D avatar.

3.2.4.1 Hair reconstruction and using hairstyles from the database

For 3D hairstyle reconstruction from a captured image, we employed the pipeline introduced in HairNet, which consists of data collection and processing, data augmentation, training data generation, and deep learning model construction. We made slight modifications to the data augmentation step to achieve better results for our specific application. During the model construction step, we utilized the improved model by Anh-Duc et al. [30], which is based on the model introduced in HairNet. We followed a similar approach to HairNet, used 343 models from open sources, and aligned all the models with the same head, using individual strand representations. Each hairstyle consisted of 32×32 fixed hair strands on the scalp grid, with each hair strand being a discrete representation of 100 points in three-dimensional space. We classified and mixed the hair dataset, resulting in a total of 2000 hairstyles. Unlike HairNet, which used large rotation angles, we used narrower rotation angles ranging from -30 to 30 degrees as our application primarily consists of front-facing images. For each hairstyle, we used three different shooting angles, which varied between distinct hairstyles.

Similar to Anh-Duc et al. [30], we added several layers to the VAE model of HairNet. In the encoder, we used residual blocks to enhance the effectiveness of working with large-size input images and multiple deep convolutional layers. We also incorporated batch normalization layers to accelerate the training speed of the model. In the final decoder step, we utilized a Conv3D block, and we added a parameter for future curliness adjustments if necessary, similar to HairNet and Anh-Duc et al. [30]. The effectiveness of these layers compared to the HairNet model is presented in the research by Anh-Duc et al. [30]. With the hair reconstruction module, our model takes in a $3 \times 256 \times 256$ image and returns a 3D hairstyle with a shape of $32 \times 32 \times 100 \times 4$.

3.3 Contribution: Customizable facial emotions

3.3.1 The idea

With a system that can take a single image input and generate a 3D avatar, we want the user to be able to have more meaningful interactions with our system. The idea is to create a simple emotion-to-FLAME regressive model to convert emotion parameters to FLAME's pose and expression parameters. The emotion-to-FLAME regressive model is trained on the VKIST dataset, with the ground truth acquired by running the dataset through the pre-trained face reconstruction model (DECA or MICA).

Two different models are trained:

- One with emotion parameters (6 parameters) as input and FLAME's pose and expression parameters as output

- One with emotion parameters (6 parameters) plus FLAME's shape parameters (50 parameters) as input and FLAME's pose and expression parameters as output

3.3.2 Model architecture

The emotion-to-FLAME regressive model is a simple feed-forward model with 6 input nodes and 56 output nodes, where the first 50 nodes are expression parameters, and the next 6 nodes are pose parameters. The model architecture is shown in the figure below.

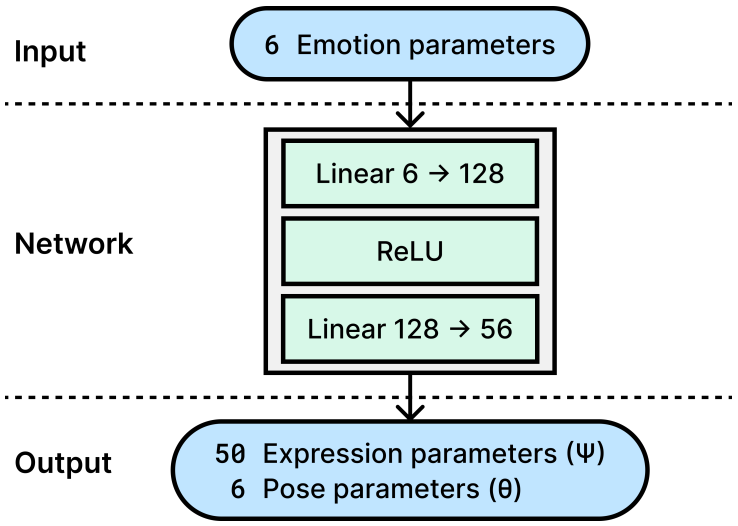

Figure 3.11: The emotion-to-FLAME model architecture.

The "linear" layers perform linear transformations on the input data, which means they apply a weight matrix and a bias vector to the input. The ReLU activation function is used to introduce non-linearity into the network. Since the emotion parameters in FLAME are projected linearly to the mesh vertices, the linear layers should perform well as a linear regressor.

The loss function used is the mean absolute error (MAE/L1) loss function L_{MAE}, which is defined as:

$$L_{MAE} = \frac{1}{N} \sum_{i=1}^{N} |y_i - \hat{y}_i| \tag{5}$$

where y_i is the ground truth, $\hat{y}_i$ is the predicted value, and N is the number of samples.

With the output of the emotion-to-FLAME model, the loss function is defined as:

$$
\begin{aligned}
Loss &= Loss(P_{exp}, \hat{P_{exp}}) + Loss(P_{pose}, \hat{P_{pose}}) \\
&= L_{MAE}(P_{exp}, \hat{P_{exp}}) + \alpha \cdot L_{MAE}(P_{neck\_pose}, \hat{P_{neck\_pose}}) \\
&\quad + \beta \cdot L_{MAE}(P_{global\_pose}, \hat{P_{global\_pose}})
\end{aligned} \tag{6}
$$

where P_{exp} is the ground truth expression parameters, $\hat{P_{exp}}$ is the predicted expression parameters, P_{pose} is the ground truth pose parameters, $\hat{P_{pose}}$ is the predicted pose parameters. Within pose parameters, there are neck pose parameters and global pose parameters. α and β are the weights for the neck pose and global pose loss, respectively.

An alternative model that additionally takes FLAME's identity parameters as input is also tested. The only difference in this model is the number of input nodes, which is 56 instead of 6. The loss function is the same as the previous model.

Chapter 4

RESULTS AND DISCUSSION

4.1 Screenshots of the system

4.2 Data Description

4.2.1 For avatar reconstruction

4.2.1.1 The HairNet dataset

We used the public HairNet dataset [26] to train the hair reconstruction model. Using our generation method, we were able to create a dataset of roughly 30,000 images for training the hair reconstruction model from the database of 343 hair models.

Figure 4.1: Rendered interpolated hairstyles in the HairNet dataset

4.2.1.2 The StyleGAN dataset

To evaluate the whole system, we used face images generated from a StyleGAN [31] model. These images are diverse in ethnicity, gender, age, and other attributes, making them suitable for evaluating the realism and accuracy of our system. Online tools such as `https://thispersondoesnotexist.com/` can be used to generate these images.

Figure 4.2: Sample images in the StyleGAN dataset.

4.2.2 For customizable facial emotions

4.2.2.1 The VKIST dataset

To train the emotion network, we use the VKIST dataset which consists of 889 images of 127 subjects with 7 captured emotions (neutral, anger, disgust, fear, happiness, sadness, surprise). All images are captured from the front view with a resolution of 2976 × 1984. The link to download the dataset is available at (`https://tinyurl.com/vkist-face-front-02`).

90% of the images are used for training and 10% for testing. From the dataset, we extract the face region and resize it to 256 × 256. We then use the pre-trained DECA model to extract the expression parameters from the face images. These expression parameters serve as the ground truth for the emotion model.

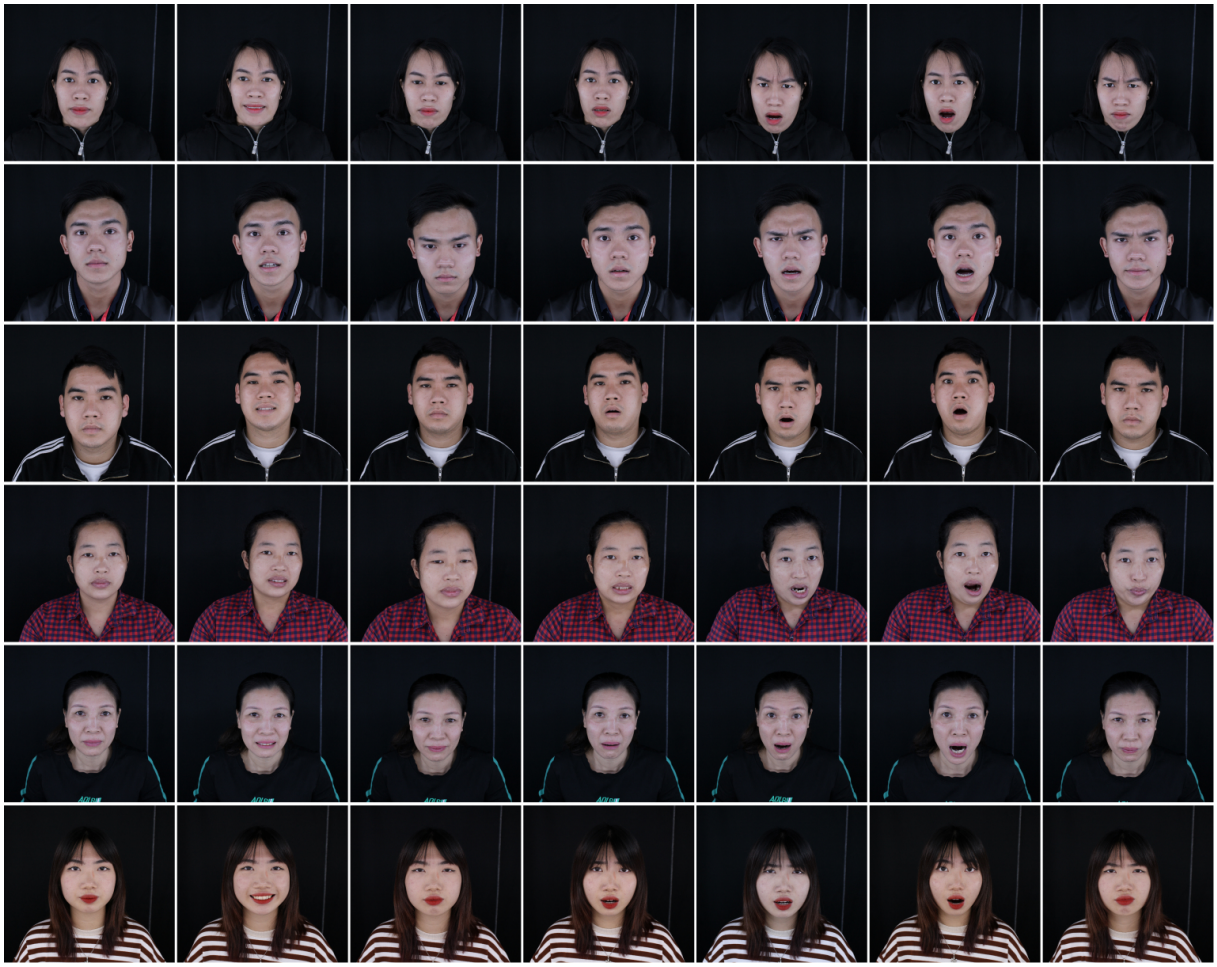

Figure 4.3: Sample images in the VKIST dataset.

4.2.2.2 The FaceScape dataset

To evaluate the emotion model, we used the FaceScape dataset, which consists of 847 subjects, is used. The dataset consists of scanned 3D face models of 20 different expressions, and 56 images corresponding to 56 camera angles to create each of the scanned models. Only the data of 359 subjects was used because the remaining subjects weren't provided with the captured images/expressions, which makes it more difficult to evaluate the effectiveness of the method.

Figure 4.4: Sample images of a subject in the FaceScape dataset.

4.3 Experimental Scenarios

To evaluate the accuracy of our system, we conducted two experiments. The first experiment is to evaluate the quality of the avatar generated by our system. The second experiment is to evaluate the quality of the customizable facial emotions.

In the first experiment, we evaluate the quality of the avatar generated by our system by comparing it with the original input image. We use the images generated by the StyleGAN model (explained in section 4.2.1.2) as the input images. We then used the HairNet-based model to reconstruct the hair and the DECA-based model to reconstruct the face. We then compared the reconstructed avatar with the original input image using several metrics which are elaborated in section 4.4.1. We also surveyed to evaluate the similarity between the original input image

and the reconstructed avatar using a Likert scale with a 5-point rating system (1-5) for qualitative evaluation.

In the second experiment, we evaluate the quality of the customizable facial emotions by comparing the reconstructed face model with the ground truth. We used the FaceScape dataset (explained in section 4.2.1.2) as the ground truth. We then used the DECA-based model to reconstruct the face with the emotion set to neutral and with the emotion set to the emotion predicted by the emotion model. We then compared the reconstructed face model with the ground truth using the method used in the NoW challenge [14].

4.4 Evaluation Methods

4.4.1 Avatar reconstruction

To evaluate the quality of the avatar generated by our system, we have designed a qualitative survey using a Likert scale with a 5-point rating system (1-5). The survey question asks respondents to rate the degree of similarity between the original input and the avatar output. This question serves as a key metric for assessing the effectiveness of our proposal. Using this question, we aimed to assess how closely the avatar output resembles the original input from the perspective of the survey participants.

4.4.1.1 Measurements: DIFD

The DIFD evaluation method determines whether two portrait images belong to the same person by comparing the difference between their embedding vectors using the Facenet model. Similar to FaceNet, we determine that two portrait images belong to the same person if their DIFD score is less than 1.5.

4.4.1.2 Measurements: PSNR, SSIM, LPIPS

PSNR and SSIM are widely used non-deep learning methods that measure similarity based on specific image attributes and provide information about the similarity in terms of noise and structure.

PSNR: Given a noise-free m×n monochrome image I and its noisy approximation K, MSE is defined as:

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2 \tag{7}$$

The PSNR is defined as

$$PSNR = 10 \cdot \log_{10} \left(\frac{MAX_I^2}{MSE} \right) \tag{8}$$

where MAX_I is the maximum possible pixel value of the image.

SSIM: The SSIM index is calculated on various windows of an image. The measure between two windows x and y of common size $N \times N$ is:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{9}$$

where μ_x is the average of x, μ_y is the average of y, σ_x^2 is the variance of x, σ_y^2 is the variance of y, and σ_{xy} is the covariance of x and y. c_1 and c_2 are two variables to stabilize the division with a weak denominator.

LPIPS: LPIPS [32] is a deep learning-based metric that employs a neural network to learn image features and compute the similarity between two images based on these features.

4.4.2 Customizable facial emotions

To evaluate the quality of the customization of the facial emotions, we follow the method used in the NoW challenge and compare the results of different method outputs with the ground truth. The ground truth is the scanned meshes of the FaceScape dataset, compared with the reconstructed face models.

4.4.2.1 Measurements: The NoW challenge

The NoW (Not quite in-the-Wild) challenge [14] provides a benchmark method specialized in measuring the accuracy and robustness of 3D face reconstruction methods. It takes 4 inputs: the ground truth mesh, the predicted mesh, the ground truth landmark points, and the predicted landmark points.

It uses a set of 7 landmark points to rigidly align the predicted mesh with the ground truth mesh. The landmark points are the leftmost and the rightmost points of the two eyes, the tip of the nose, the leftmost point, and the rightmost point of the mouth. The error is then calculated using the absolute distance between each scan vertex and the closest point in the mesh surface. The error is output as a vector of error values, in milliter units, for each vertex in the mesh. The average, median, and standard deviation of the error vector are then calculated and output as the final result.

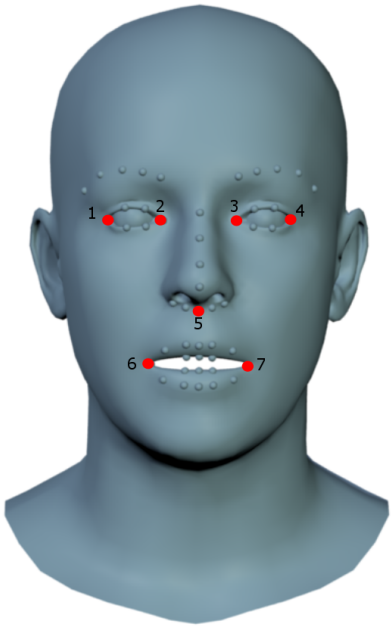

Figure 4.5: The landmark points used in the NoW challenge.

Figure 4.6: The area used for evaluation in the NoW challenge.

4.5 Experimental Results and Commentary

4.5.1 Avatar reconstruction

4.5.1.1 Quantitative results

Table 4.1 illustrates the results of the aforementioned measurements when comparing our method with several other 3D face reconstruction methods. The results show that, in terms of comparison, our results are not as good as many other methods such as i3DMM and MoFaNeRF because they applied the measurements to hairless faces. However, we also see that all of the measurement results meet the requirements. In particular, the average value of DIFD is 0.25, which indicates that the synthesized output image has been evaluated as retaining the represented features of the same person as the input image.

Table 4.1: Comparison of our proposal and others

	PSNR	**SSIM**	**LPIPS**	**DIFD**
Our system	23.15	0.835	0.09	0.25
i3DMM	24.45	0.904	0.11	NA
MoFaNeRF	31.49	0.951	0.06	NA

4.5.1.2 Qualitative results

Out of the 33 respondents, the survey showed that an impressive 93.8% of the respondents were able to correctly identify that the input image and the synthesized face image belonged to the same ($score >= 3$). Of those, 14% were evaluated as being very similar with scores of 5, and 49.3% were evaluated with scores of 4.

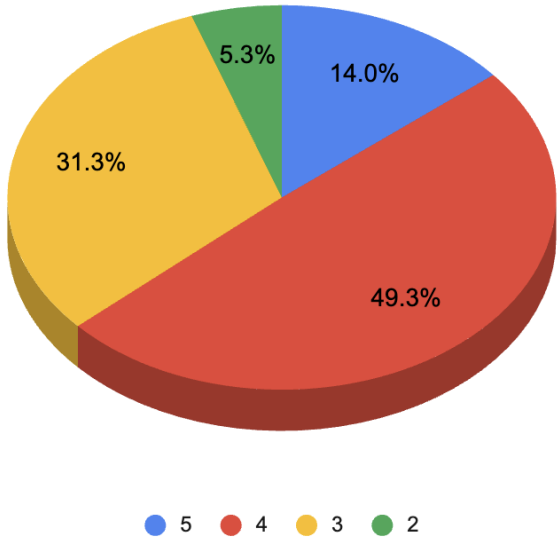

Figure 4.7: The survey result.

4.5.2 Customizable facial emotions

4.5.2.1 Quantitative results

We use the FaceScape scanned mesh dataset as the ground truth for the evaluation. The comparison is made between the ground truth and the reconstructed mesh. We compare the ground truth with 3 types of face reconstruction models: One uses the DECA model with the emotion set to neutral, and one uses DECA with auto-detected emotion. One uses the DECA model with our emotion model. The comparison is additionally made with two views using different input images: one from the front view and one from the side view. The results are shown in the table below. The benchmarking method is the NoW challenge [14], which is the most widely used method for evaluating 3D face reconstruction methods.

Table 4.2: Comparison of our emotion model with others

Method	Front view image input			Side view image input		
	Mean	Median	Std	Mean	Median	Std
DECA + neutral emotion						
DECA + detected emotion						
DECA + our model						

Chapter 5

CONCLUSIONS

5.1 Conclusions

REFERENCES

[1] 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*, Not Known: ACM Press, 1999, pp. 187–194, ISBN: 978-0-201-48560-8. DOI: 10.1145/311535.311556. [Online]. Available: http://portal.acm.org/citation.cfm?doid=311535.311556 (visited on 06/05/2023).

[2] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter, "A 3D Face Model for Pose and Illumination Invariant Face Recognition," in *2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*, Sep. 2009, pp. 296–301. DOI: 10.1109/AVSS.2009.58.

[3] T. Gerig, A. Morel-Forster, C. Blumer, *et al.*, "Morphable Face Models - An Open Framework," in *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, May 2018, pp. 75–82. DOI: 10.1109/FG.2018.00021.

[4] C. Cao, Y. Weng, S. Zhou, Y. Tong, and K. Zhou, "FaceWarehouse: A 3D Facial Expression Database for Visual Computing," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 3, pp. 413–425, Mar. 2014, ISSN: 1941-0506. DOI: 10.1109/TVCG.2013.249.

[5] 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*, vol. 36, no. 6, 194:1–194:17, Nov. 20, 2017, ISSN: 0730-0301. DOI: 10.1145/3130800.3130813. [Online]. Available: https://dl.acm.org/doi/10.1145/3130800.3130813 (visited on 06/05/2023).

[6] H. Yang, H. Zhu, Y. Wang, *et al.*, "FaceScape: A Large-Scale High Quality 3D Face Dataset and Detailed Riggable 3D Face Prediction," presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 601–610. [Online]. Available: https://openaccess.thecvf.com/content_CVPR_2020/html/Yang_FaceScape_A_Large-Scale_High_Quality_3D_Face_Dataset_and_Detailed_CVPR_2020_paper.html (visited on 06/13/2023).

[7] A. Ranjan, T. Bolkart, S. Sanyal, and M. J. Black, "Generating 3D Faces using Convolutional Mesh Autoencoders," presented at the Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 704–720. [Online]. Available: https://openaccess.thecvf.com/content_ECCV_2018/html/Anurag_Ranjan_Generating_3D_Faces_ECCV_2018_paper.html (visited on 06/13/2023).

[8] B. Gecer, S. Ploumpis, I. Kotsia, and S. Zafeiriou, "Fast-GANFIT: Generative Adversarial Network for High Fidelity 3D Face Reconstruction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 9, pp. 4879–4893, Sep. 2022, ISSN: 1939-3539. DOI: 10.1109/TPAMI.2021.3084524.

[9] S. Galanakis, B. Gecer, A. Lattas, and S. Zafeiriou, "3DMM-RF: Convolutional Radiance Fields for 3D Face Modeling," presented at the Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2023, pp. 3536–3547. [Online]. Available: https://openaccess.thecvf.com/content/WACV2023/html/Galanakis_3DMM-RF_Convolutional_Radiance_Fields_for_3D_Face_Modeling_WACV_2023_paper.html (visited on 06/13/2023).

[10] Y. Hong, B. Peng, H. Xiao, L. Liu, and J. Zhang, "HeadNeRF: A Real-Time NeRF-Based Parametric Head Model," presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 20 374–20 384. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2022/html/Hong_HeadNeRF_A_Real-Time_NeRF-Based_Parametric_Head_Model_CVPR_2022_paper.html (visited on 06/13/2023).

[11] Y. Feng, H. Feng, M. J. Black, and T. Bolkart, "Learning an animatable detailed 3D face model from in-the-wild images," *ACM Transactions on Graphics*, vol. 40, no. 4, 88:1–88:13, Jul. 19, 2021, ISSN: 0730-0301. DOI: 10.1145/3450626.3459936. [Online]. Available: https://dl.acm.org/doi/10.1145/3450626.3459936 (visited on 05/28/2023).

[12] W. Zielonka, T. Bolkart, and J. Thies. "Towards Metrical Reconstruction of Human Faces." arXiv: 2204.06607 [cs]. (Oct. 19, 2022), [Online]. Available: http://arxiv.org/abs/2204.06607 (visited on 06/01/2023), preprint.

[13] J. Deng, J. Guo, T. Liu, M. Gong, and S. Zafeiriou, "Sub-center ArcFace: Boosting Face Recognition by Large-Scale Noisy Web Faces," in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., vol. 12356, Cham: Springer International Publishing, 2020, pp. 741–757, ISBN: 978-3-030-58620-1 978-3-030-58621-8. DOI: 10.1007/978-3-030-58621-8_43. [Online]. Available: https://link.springer.com/10.1007/978-3-030-58621-8_43 (visited on 11/22/2023).

[14] S. Sanyal, T. Bolkart, H. Feng, and M. Black, "Learning to regress 3D face shape and expression from an image without 3D supervision," in *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019, pp. 7763–7772.

[15] K. Ward, F. Bertails, T.-y. Kim, S. R. Marschner, M.-p. Cani, and M. C. Lin, "A Survey on Hair Modeling: Styling, Simulation, and Rendering," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 2, pp. 213–234, Mar. 2007, ISSN: 1941-0506. DOI: 10.1109/TVCG.2007.30. [Online]. Available: https://ieeexplore.ieee.org/document/4069232 (visited on 11/23/2023).

[16] J. Deng, J. Guo, Y. Zhou, J. Yu, I. Kotsia, and S. Zafeiriou. "RetinaFace: Single-stage Dense Face Localisation in the Wild." arXiv: 1905.00641 [cs]. (May 4, 2019), [Online]. Available: http://arxiv.org/abs/1905.00641 (visited on 11/23/2023), preprint.

[17] K. Papadopoulos, A. Kacem, A. Shabayek, and D. Aouada. "Face-GCN: A Graph Convolutional Network for 3D Dynamic Face Identification/Recognition." arXiv: 2104.09145 [cs]. (Apr. 20, 2021), [Online]. Available: http://arxiv.org/abs/2104.09145 (visited on 11/23/2023), preprint.

[18] S. Saito, Z. Huang, R. Natsume, S. Morishima, A. Kanazawa, and H. Li. "PIFu: Pixel-Aligned Implicit Function for High-Resolution Clothed Human Digitization." arXiv: 1905.05172 [cs]. (Dec. 3, 2019), [Online]. Available: http://arxiv.org/abs/1905.05172 (visited on 11/23/2023), preprint.

[19] W. Zeng, W. Ouyang, P. Luo, W. Liu, and X. Wang, "3D Human Mesh Regression With Dense Correspondence," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA: IEEE, Jun. 2020, pp. 7052–7061, ISBN: 978-1-72817-168-5. DOI: 10.1109/CVPR42600.2020.00708. [Online]. Available: https://ieeexplore.ieee.org/document/9157190/ (visited on 11/23/2023).

[20] X. D. Yang, Z. Xu, J. Yang, and T. Wang, "The Cluster Hair Model," *Graphical Models*, vol. 62, no. 2, pp. 85–103, Mar. 1, 2000, ISSN: 1524-0703. DOI: 10.1006/gmod.1999.0518. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1524070399905180 (visited on 11/23/2023).

[21] Y. Bao and Y. Qi, "A Survey of Image-Based Techniques for Hair Modeling," *IEEE Access*, vol. 6, pp. 18 670–18 684, 2018, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2018.2818795. [Online]. Available: http://ieeexplore.ieee.org/document/8323371/ (visited on 11/23/2023).

[22] M. Zhang, M. Chai, H. Wu, H. Yang, and K. Zhou, "A data-driven approach to four-view image-based hair modeling," *ACM Transactions on Graphics*, vol. 36, no. 4, 156:1–156:11, Jul. 20, 2017, ISSN: 0730-0301. DOI: 10.1145/3072959.3073627. [Online]. Available: https://doi.org/10.1145/3072959.3073627 (visited on 11/23/2023).

[23] M. Chai, L. Wang, Y. Weng, X. Jin, and K. Zhou, "Dynamic hair manipulation in images and videos," *ACM Transactions on Graphics*, vol. 32, no. 4, 75:1–75:8, Jul. 21, 2013, ISSN: 0730-0301. DOI: 10.1145/2461912.2461990. [Online]. Available: https://doi.org/10.1145/2461912.2461990 (visited on 11/23/2023).

[24] M. Chai, L. Wang, Y. Weng, Y. Yu, B. Guo, and K. Zhou, "Single-view hair modeling for portrait manipulation," *ACM Transactions on Graphics*, vol. 31, no. 4, 116:1–116:8, Jul. 1, 2012, ISSN: 0730-0301. DOI: 10.1145/2185520.2185612. [Online]. Available: https://doi.org/10.1145/2185520.2185612 (visited on 11/23/2023).

[25] L. Hu, C. Ma, L. Luo, and H. Li, "Single-view hair modeling using a hairstyle database," *ACM Transactions on Graphics*, vol. 34, no. 4, 125:1–125:9, Jul. 27, 2015, ISSN: 0730-0301. DOI: 10.1145/2766931. [Online]. Available: https://doi.org/10.1145/2766931 (visited on 11/23/2023).

[26] Y. Zhou, L. Hu, J. Xing, *et al.*, "HairNet: Single-View Hair Reconstruction Using Convolutional Neural Networks," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2018, pp. 249–265, ISBN: 978-3-030-01252-6. DOI: 10.1007/978-3-030-01252-6_15.

[27] H. Feng, *Photometric FLAME Fitting*, Jun. 14, 2023. [Online]. Available: https://github.com/HavenFeng/photometric_optimization (visited on 06/14/2023).

[28] W. A. P. Smith, A. Seck, H. Dee, B. Tiddeman, J. B. Tenenbaum, and B. Egger, "A Morphable Face Albedo Model," presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 5011–5020. [Online]. Available: https://openaccess.thecvf.com/content_CVPR_2020/html/Smith_A_Morphable_Face_Albedo_Model_CVPR_2020_paper.html (visited on 06/14/2023).

[29] S. Xu, J. Yang, D. Chen, *et al.*, "Deep 3D Portrait From a Single Image," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA: IEEE, Jun. 2020, pp. 7707–7717, ISBN: 978-1-72817-168-5. DOI: 10.1109/CVPR42600.2020.00773. [Online]. Available: https://ieeexplore.ieee.org/document/9156419/ (visited on 06/10/2023).

[30] A. D. Lo and T. C. Ma, "Three-Dimensional Hair Structure Reconstruction from a Single Sketch Image without Intermediate Representation," Vietnam-Korea University of Information and Communication Technology, Working Paper, Jun. 2023. [Online]. Available: http://elib.vku.udn.vn/handle/123456789/2700 (visited on 11/23/2023).

[31] T. Karras, S. Laine, and T. Aila, "A Style-Based Generator Architecture for Generative Adversarial Networks," presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 4401–4410. [Online]. Available: https://openaccess.thecvf.com/content_CVPR_2019/html/Karras_A_Style-Based_Generator_Architecture_for_Generative_Adversarial_Networks_CVPR_2019_paper.html (visited on 06/14/2023).

[32] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. "The Unreasonable Effectiveness of Deep Features as a Perceptual Metric." arXiv: 1801.03924 [cs]. (Apr. 10, 2018), [Online]. Available: http://arxiv.org/abs/1801.03924 (visited on 11/22/2023), preprint.