이화여자대학교 대학원
2020 학년도
석사학위 청구논문

Facial Geometry Synthesis Using Locally Weighted Autoencoder

인 공 지 능 · 소 프 트 웨 어 학 부
Minyoung Kim

2021

Facial Geometry Synthesis Using Locally Weighted Autoencoder

이 논문을 석사학위 논문으로 제출함

2021 년 12 월

이화여자대학교 대학원

인공지능 · 소프트웨어학부 Minyoung Kim

Minyoung Kim의 석사학위 논문을 인준함

지도교수 김 영 준 ____

심사위원 민 동 보 ____

오 유 란 ____

김 영 준 ____

이화여자대학교 대학원

Table of Contents

List of Figures

List of Tables

Abstract

Various methods have been studied to develop three-dimensional(3D) geometric models to generate human faces in three dimensions. With the advent of the generative adversarial networks (GAN), attention to the generation model is increasing, and related research using deep learning is being actively progressed. Face generation can be utilized to create virtual avatars and virtual human content that has recently emerged.

In this dissertation, we proposed a three-dimensional face generative model with local weights to increase the model's variations and expressiveness. Previous studies on face generative models have attempted to create a variety of faces by using entire large faces. However, dividing the face into several parts with semantic features can improve the model's representation ability and addresses the limitations of insufficient datasets.

Unlike previous studies that required learning of the entire face mesh, and where part manipulation is impossible, the proposed model allows partial manipulation of the face while still learning the whole face mesh. For this purpose, we address the identification of an effective way to extract local facial features from the entire data and explore a way to enable a holistic generation by learning local features. By factorizing the latent vector of the whole face, latent vectors from the subspace can be used to indicate different parts of the face. In addition, local weights generated by non-negative matrix factorization (NMF) are applied to the factorized latent space so that the decomposed part space is more semantically meaningful.

We experiment with the proposed model and observe that effective facial part manipulation is possible, and the model's expressiveness is improved. In addition, several ablation tests have shown that the local weights proposed in this study produce meaningful results.

I. Introduction

A. Motivation

The representation and synthesis of the 3D human face form one of the active research topics in computer graphics and computer vision in such applications as face recognition [1], reconstruction [2], generation [3], and animation [4]. Its importance has increased lately due to the development of virtual reality, especially virtual humans [5]. However, modeling a human face still needs a tremendous human effort. Many researchers have proposed new approaches to address this difficulty. Among them, the learning-based method exhibits notable advancements.

Generative adversarial networks (GANs) have shown realistic results in generating a human face image and have also spurred research to generate 3D human faces with deep neural networks. Many existing works represent a human face with various types of 3D representations, e.g., mesh [6], voxel [7], point cloud [8], and geometry image [9]. These show state-of-the-art performances such as photo-realistic face texture and fine-detailed geometry model.

To represent a 3D human face, the triangular mesh has been a favorite in several research works because of its efficiency, non-uniformity of representation, and scalability for other applications. In contrast, voxel requires high computational performance, and the point cloud has an absence of smoothness of the data representation. Geometry image ([3], [9]) presents high-quality face generation enabling photo-realistic face synthesis. However, it is also limited to the uniformity of 2D image pixels.

The advance of deep neural networks has been influential to computer graphics methodologies and has influenced geometry processing. Geometric deep learning attempts to generalize neural

networks and apply convolutional neural networks (CNNs) to graphs and manifolds [10]. There exists much research applying CNNs to graph structures ([11], [12]) and meshes ([13], [6], [14]).

With the advantages of CNNs for hierarchical feature extraction, many generative models also capitalize on its benefits to progress shape modeling ([15], [16], [17]). However, most existing works are focused on a holistic generative approach, which is constrained to generate all parts at once and lacks part details and manipulation.

Previous studies have mentioned the benefit of a localized model and part-based representation. Blanz and Vetter [18] say that independently morphed parts of faces can intensify the expressiveness of the model. Tena et al. [19] also assert that applying segmentation to the face model allows for the generation of combined facial parts beyond those in the training data. Finally, Tran et al. [20] suggest that local models are not only more expressive than global models but are also less expensive to represent human faces realistically. Following these previous discoveries, it would be worth exploring a local method applying deep neural networks.

In other aspects, the representation ability of the learning-based generative model depends on the volume and quality of a 3D face dataset for training. However, the scarcity of 3D face training data is indeed one of the challenges in 3D face generative model research. Unlike 2D face datasets ([21], [22]), 3D face datasets barely include segmentation labels and data, and this is not an easy task to achieve.

B. Research Goal

The goal of this study is to synthesize a 3D human face with generative power. In this dissertation, we propose a locally weighted 3D face generative model to increase variations and expressiveness of the model. We expect that our approach can generate a rich variety of 3D face

models beyond the training data using part manipulation with latent factorization. With a part-based representation of the data, our model is simpler and more straightforward than others and does not require any semantic segmentation labels.

To achieve these goals, we need to address several challenges. First, we must investigate an effective way to extract or present localized features from the whole data. Mesh segmentation is one of the possible solutions. However, since human faces are often smooth, it is a challenge to segment the facial mesh explicitly. To bypass this, we exploit a holistic generative approach that does not require additional segmentation data and makes the whole learning model simple. Furthermore, we explore a way for part control while using holistic generation by learning localized features.

C. Main Contributions

Our main contributions are as follows:

- Locally weighted generative autoencoder for generating a whole human face geometric model.
- End-to-end learning to learn local features without explicit facial feature segmentation data.
- Experimentation and demonstration of the proposed model's performance in terms of generation, reconstruction, and part manipulation.

We utilize Ranjan et al. [23]'s autoencoder with latent space factorization and apply local weights that partially influence the model during training. Latent factorization enables manipulation of the local part of the face, and local weights make decomposed part spaces more semantically meaningful without additional segmentation labels due to its part-based representation. We also evaluate the performance of the proposed model in terms of reconstruction ability, part modification,

part combination, and several ablation tests to show the effect of each model component on the results.

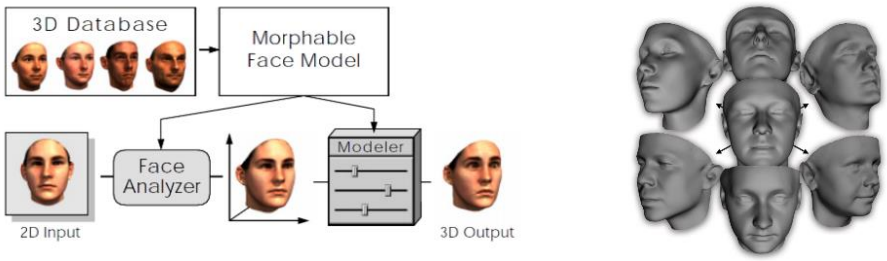
D. Organization

The remainder of this dissertation is organized as follows. Section II presents the previous research on 3D face representation and generative model, part-based shape generative model, and feature matrix decomposition. In Section III, we briefly explain the convolution neural networks used in our model. Section IV explains our proposed model in detail and Section V provides its implementation details. Section VI presents the experiments and the visualization of the results. Finally, we give the conclusions in Section VII.

II. Related Work

A. 3D Face Representations and Generative Models

Three-dimensional face shape modeling must present geometric facial shapes with variations across diverse identities and expressions [24]. Blanz and Vetter [18] introduced the first 3D face *morphable models (3DMMs)*, which are statistical models of global 3D face shapes and textures. They employed principal component analysis (PCA) to construct principal components to express facial shape and texture (Figure 1-(a)). More recently, Booth et al. proposed the first largest scale *morphable model*, the large scale face model (LSFM) [25], constructed from 9663 distinct facial identities, as shown in Figure 1-(b). Paysan et al.'s Basel face model (BFM) [26] also has been widely used. However, the 3DMMs are limited to the representation of high-frequency details and form a latent model space due to their linear bases and training data.



(a) 3D *morphable models* [18] (b) Large scale face model (LSFM) [25]

Figure 1 PCA-based face models

Recently, the development of deep learning provided a new approach to generate 3D shapes with non-linear parametric models. Fernandez et al. [27] suggest the first autoencoder with a CNN-based encoder and a tensor-based face model as a decoder for generating 3D face shapes. Similar to [27], various researchers ([28], [29], [3], [9]) have exploited the 2D representation of geometry image due to the difficulty of applying convolution to the 3D mesh directly (Figure 2). In order to overcome these difficulties, Ranjan et al. [23] proposed the first autoencoder architecture, convolutional mesh autoencoders (CoMA), that performs 3D convolutions with truncated Chebyshev poynomials [11] applied directly to the 3D mesh (Figure 3). MeshGAN [30] is a combination of GANs with truncated Chebyshev poynomials [11]. Extending [23], Li et al. [31] suggest multi-column graph convolution networks that applying a different Chebyshev convolution filter scale.

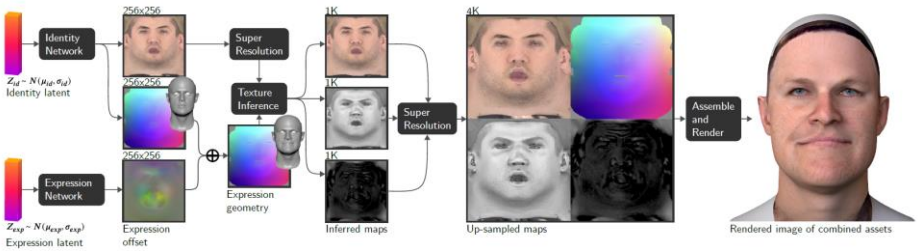


Figure 2 Overview of the generative model using geometry images in [9]

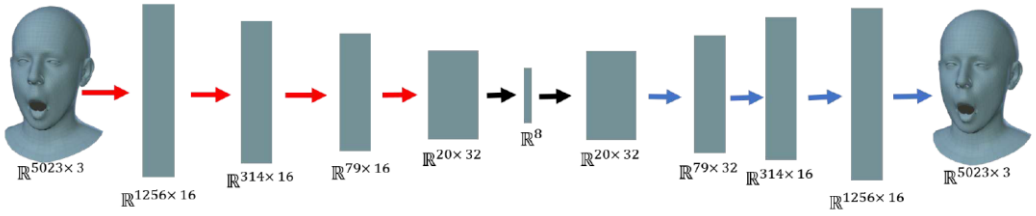
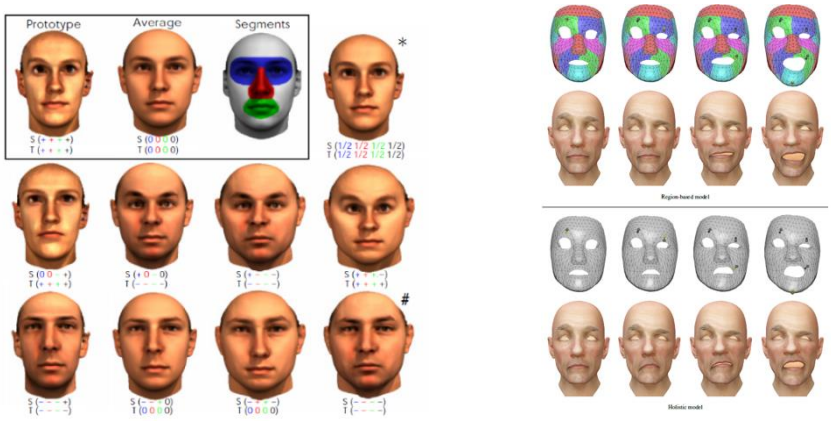


Figure 3 Convolutional mesh autoencoder [23]

Despite their performance, all these learning-based methods are labeled as *holistic* generation that generates all parts at once. For this reason, the above studies had difficulties in manipulating local anatomical parts such as eyes, nose, and mouth. They also had difficulty in generating a fine-scale geometric model.



(a) The morphable model [18] added a large partial variety of faces

(b) Region-based model and holistic model from Tena et al. [19]

Figure 4 Part-based approach applied to face generation

There exist attempts to generate a new face with face segmentation and a local model to increase the model's expressiveness and achieve fine-scale modeling. Blanz and Vetter [18] demonstrate region-based modeling with 3DMMs by manually dividing the face into regions that can be learned by the PCA models (Figure 3-(a)). Tena et al. [19] present region-based linear face modeling with automatic segmentation by clustering (Figure 4-(b)). Tran et al. [20] also propose non-linear 3DMMs with a global and local-based network to extract features of the global face structure and face part details simultaneously. Recently, Ghafourzadeh et al. [32] proposed a part-

based approach that conducts part-based facial models using PCA. This model results in a locally edited face by applying an anthropometric measurement.

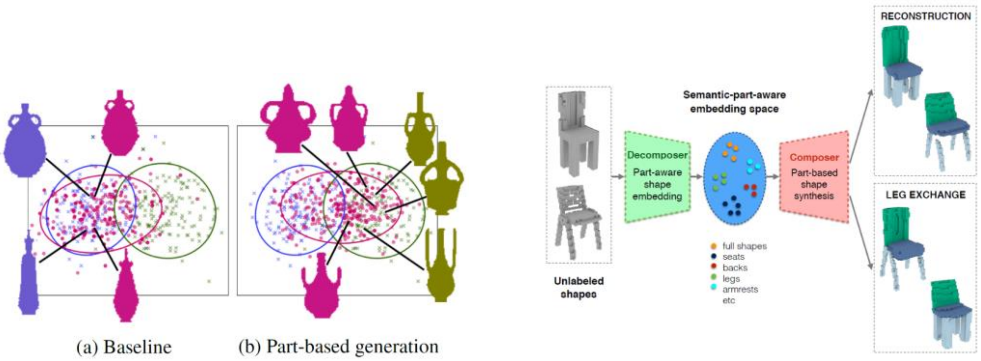
Nevertheless, to the best of our knowledge, a learning-based localized generation model that generates face data has not been proposed yet.

B. Part-Based Shape Generative Models

Wang et al. [16] propose holistic voxel-based generative adversarial networks called global-to-local GAN and part refiner. At first, global-to-local GAN generates the whole shape and segments it into several parts. Then it refines the rough parts of the object with upsampling from low resolution to high resolution. They show better shape variety and distribution than a plain three-dimensional GAN by training the data with 3D inception score measurements.

CompoNet [33] presents a part-based generative neural network for shapes. It suggests two units: the part synthesis unit and the part composition unit. The synthesis unit consists of parallel generative autoencoders that learn each semantic part of the shape. The composition unit learns to compose the encoded parts. With the suggested model, Schor et al. [33] proved that the part-based model encourages the generator to create new data that was unseen in the training set.

Dubrovina et al. [34] handle the composition and decomposition of each part as a simple linear operation on the factorized embedding space, reflecting the part structure and encoding the geometry of the different semantic parts. They use projection matrices to split full object encodings into part encodings and represent them as fully connected layers. They show that the proposed decomposer-composer network can perform meaningful part manipulations and high-fidelity 3D shape generation.



(a) CompoNet [33] (b) Decomposer-composer [34]

Figure 5 Part-based shape generative models

To composite each part, both [33] and [34] compute per-part affine transformation. This task requires ground truth data of the parts. Our model does not utilize spatial transformer networks [35] nor computes affine transformation to combinate each part of the data.

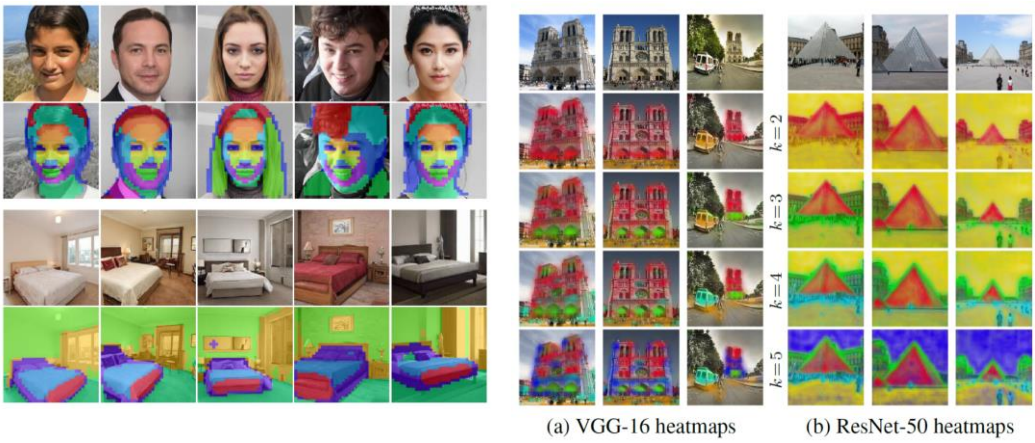
Öngün and Temizel [36] propose a holistic approach to learning the semantic part of the autoencoder data. They can handle part editing and modification without additional part assembly. However, they use a part-segmented point cloud dataset.

We pursue a holistic generation approach but also allow part manipulation without explicit segmentation. Therefore, we do not need to worry about the artifacts when the model combines each part into a whole shape.

C. Feature Matrix Factorization

Some feature factorization methods interpret data more semantically since they can decompose the data into a part-based representation. Collins et al. [37] perform local and semantically-aware

changes through a global operation on the 2D image domain. They apply spherical k-means clustering [38] on the last feature map to identify features that are semantically meaningful (Figure 6-(a)). Deep feature factorization [39] influenced their research. Collins and Süsstrunk [39] demonstrate localized features using non-negative matrix factorization (NMF) (Figure 6-(b)). They apply NMF to the last feature map, where the semantic features are encoded. By factorizing the feature map, they can decompose an input image into several semantic regions.



(a) *k* heatmap by *k*-means clustering [38]

(b) *k* heatmap generated by Deep Feature Factorization (DFF) [39]

Figure 6 Feature decomposition examples from [37] and [39]

NMF is a robust feature factorization method to represent data as part based. Lee and Seung [40] popularized NMF by showing its interpretability for part-based representation of facial images. Koppen et al. [41] extended NMF to 3D registered images. McGraw et al. [42] present 3D segmentation based on NMF and produce meaningful results. For its application, Li et al. [43] propose the concept of *sketch* as an input of GANs, which is the noise transformed to the basis matrix in NMF that has the underlying features of the raw data.

By applying NMF to 3D faces mesh, we supply localized weights to the holistic generative model.

III. Mesh Convolution Neural Networks

In this study, we choose to represent 3D faces with triangular mesh due to its efficiency. Additionally, we utilize a learning-based model using CNNs. However, applying convolution neural networks to 3D meshes is not as straightforward as to images. Among diverse approaches, Ranjan et al. [23] propose CoMA employing fast Chebyshev filters [11] with a novel mesh pooling method. With their model, Ranjan et al. [23] show outstanding performance on 3D face reconstruction and learning a non-linear representation.

There are other networks for a mesh that we attempt to employ: Bouritsas et al. [6]'s spiral convolution networks and Hanocka et al. [13]'s MeshCNN. Both studies perform state-of-the-art 3D mesh representation, but they are not suitable for our research purpose for the following reasons. First, spiral convolution networks need large pre-computed spiral trajectory data that is inappropriate for our high-resolution face data. It also requires an increased training time. Next, MeshCNN [13] cannot perform arbitrary mesh generation tasks since their algorithm requires the previous downsampling history. It means that the model using MeshCNN is unable to run a decoder or generator only.

Therefore, we employ CoMA [23] to learn 3D facial mesh data and generate a new mesh because of its stable performance, adequate training time, and scalability of the model. Since our model is largely based on this model, we briefly explain the spectral graph convolution the model used.

A. Mesh Representation

We represent a 3D face mesh as a set of vertices $V \in \mathbb{R}^{N\times 3}$ and edges A. The edges are represented by an adjacency matrix $A \in \{0,1\}^{N\times N}$, where $A_{ij} = 1$ denotes where there is an edge connecting vertices i and j, and $A_{ij} = 0$ otherwise.

B. Spectral Graph Convolution on Face Mesh

Defferrard et al. [11] use convolution on graphs with a frequency domain approach under the convolution theorem. The convolution in the spatial domain equals element-wise multiplication in the frequency domain. To convert the graph from the spatial domain to the frequency domain, Defferrard et al. [11] first applied the graph Fourier transform [15] to the input mesh. The graph Laplacian matrix is defined as $L = D - A$, where D is a diagonal matrix, with $D_{i,i} = \Sigma_j A_{ij}$. The Laplacian matrix is diagonalized by the Fourier basis $U \in \mathbb{R}^{N\times N}$ as $L = U\Lambda U^T$. Here, the columns of $U = [u_0, u_1, \dots, u_{n-1}]$ are the orthogonal eigenvectors of L, and $\Lambda = diag([\lambda_0, \lambda_1, \dots, \lambda_{n-1}]) \in \mathbb{R}^{N\times N}$ is a diagonal matrix. Following the convolution theorem, the convolution operator $*$ can be defined in the Fourier space as the element-wise product $X * W_{spec} = U(U^T(X) \odot U^T(W_{spec}))$. Because of U, which is not sparse, this operation needs high computational costs. To address this problem, Defferrard et al. [11] formulate spectral convolution with a filter W_θ using a recursive Chebyshev polynomial ([11], [44]). The filter W_θ is parametrized as a Chebyshev polynomial of order K by

$$W_\theta(L) = \sum_{k=0}^{K-1} \theta_k T_k(\tilde{L}), \tag{1}$$

where $\tilde{L} = \frac{2L}{\lambda_{max}} - I_n$ is the scaled Laplacian matrix, and λ_{max} is the maximum eigenvalue of the Laplacian matrix. The parameter $\theta \in \mathbb{R}^K$ is a vector of the Chebyshev coefficients, and $T_k \in \mathbb{R}^{N \times N}$ is the Chebyshev polynomial of order k, which is computed recursively as $T_k(x) = 2xT_{k-1}(x) - T_{k-1}(x)$, with $T_0 = 1$ and $T_1 = x$. For each convolution layer, the spectral graph convolutions are

$$\mathcal{Y}_j = \sum_{i=0}^{F_{in}} W_{\theta_{i,j}}(L)\, x_i \tag{2}$$

where x_i is the i-th feature of the input $x \in \mathbb{R}^{N \times F_{in}}$, and $\mathcal{Y}_j$ is the j-th feature of the output $\mathcal{Y} \in \mathbb{R}^{N \times F_{out}}$. For each convolution layer, the spectral graph convolution has $F_{in} \times F_{out}$ vectors of the Chebyshev coefficient $\theta_{i,j} \in \mathbb{R}^K$ as trainable parameters.

C. Mesh Sampling

The hierarchical operation allows CNNs to learn global and local features, one of the networks' strong advantages. It requires downsampling and upsampling operations to reduce the data's dimensions and make them coarse and fine. For this hierarchical multiscale representation, Ranjan et al. [23] introduced mesh sampling operations to capture both global and local contexts. This

technique defines a new topological structure of each downsampling and upsampling layer and maintains the context on neighborhood vertices.

In downsampling, removed vertices are selected using the quadric error metric The technique stores the barycentric location of the removed vertices w.r.t. what remains. The downsampled mesh passes through convolutional operations. Finally, removed vertices are added to the stored barycentric locations (Figure 7).

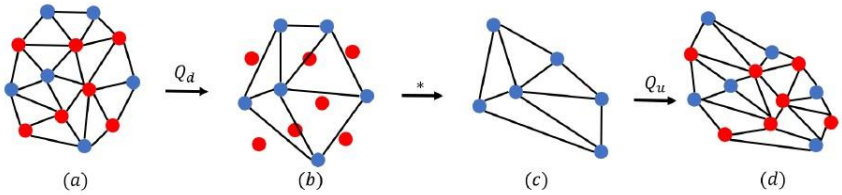


Figure 7 Mesh sampling operation [23]

IV. Locally Weighted Autoencoder

A. Overview

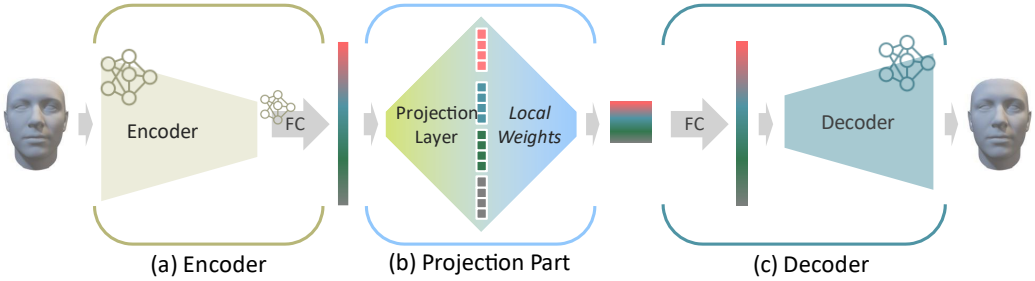


Figure 8 Locally weighted autoencoder (simplified)

Our model aims to synthesize various 3D faces and manipulate them locally while generating faces holistically. Our model is based on an autoencoder, one of the unsupervised learning techniques. It consists of three parts: an encoder (Figure 8-(a)), projection (Figure 8-(b)), and a decoder (Figure 8-(c)). Figure 8 shows an overview of the model architecture.

The encoder mainly compresses and encodes the input data. It takes face meshes as input and encodes them to low-dimension vectors using the convolution operation and a fully connected layer during training. Next, the projection part divides the latent space into several subspaces and makes them semantically meaningful. The latent space is factorized by learnable projection matrices. Thus, the factorized latent space reflects the part structure of the shape, not the whole structure. The latent vectors from the factorized space are then multiplied by local weights, computed before training the model. We produce local weights by using NMF to represent the data's whole structure as semantic

part structures. The encodings passed through the projection part correspond to the shapes' semantic part structure and are known as locally weighted encodings.

Lastly, the decoder decompresses the sum of these encodings, which means that it reconstructs the latent vector to the original input, i.e., the face. The whole model repeats this process and eventually learns how to reconstruct and form the data's latent space. Figure 9 describes the detailed model architecture.

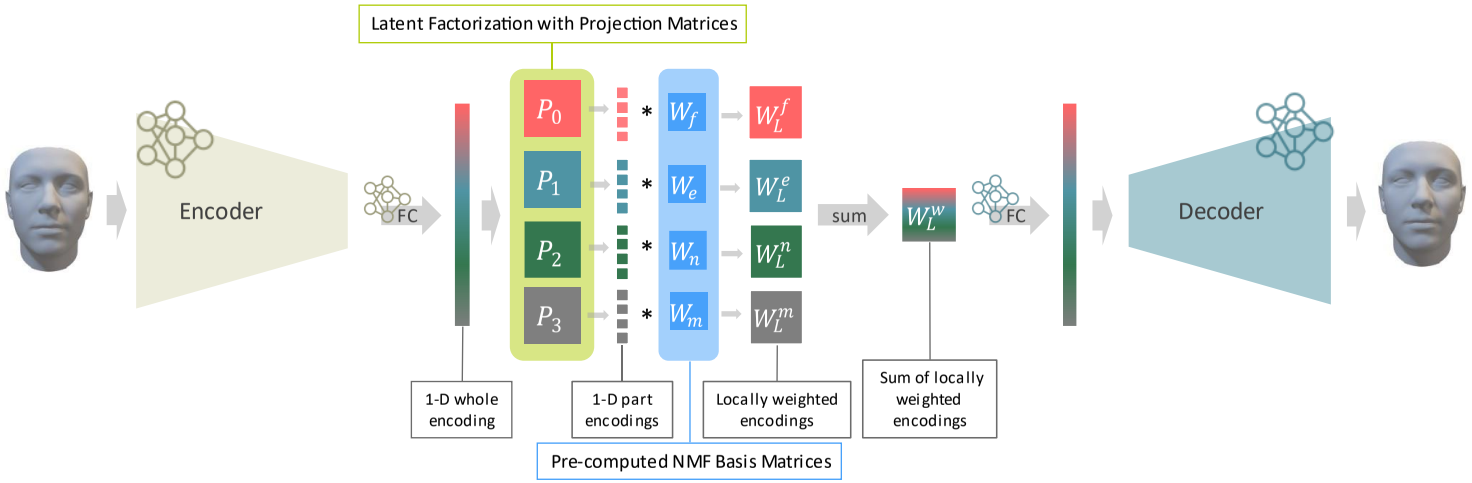
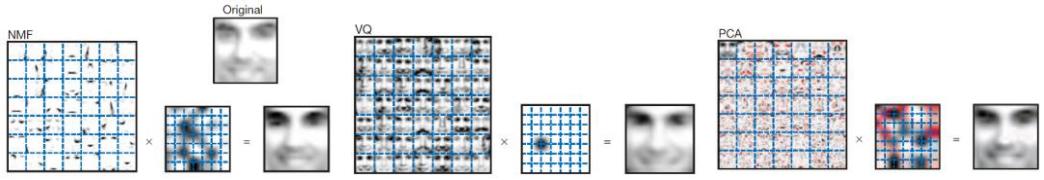


Figure 9 Locally weighted autoencoder architecture overview

B. Pre-Computed Local Weights from NMF

In this section, we explain local weights that have semantic meanings presented by parts-based representation. Most previous parts-based generative models exploit explicit segmentation data or label for training their model to learn the structure of the object parts. However, 3D facial mesh datasets barely have those pre-segmented data. Contrary to the previous studies, we use parts-based representation to extract the local part structure without segmented data or labels. We refer to the represented data as local weights, which have each vertex's influence on each divided facial part. We employ NMF to present the parts-based representation of the whole data.

NMF is a linear dimensionality reduction technique. It learns a part-based representation of the data, as opposed to other methods, such as vector quantization (VQ) and principal components analysis (PCA), that learn a holistic representation (Figure 10). Lee and Seung [40] explain that the non-negativity constraints allow only additive, not subtractive, combinations. For these reasons, NMF is used for mesh segmentation [42], document clustering [45], and other applications.



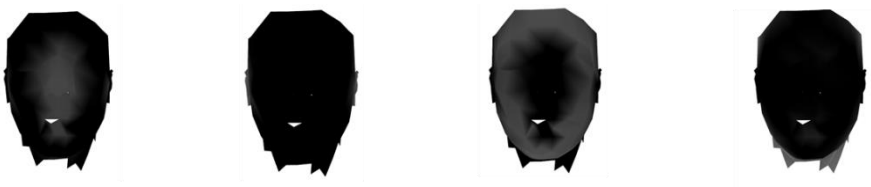
(a) NMF (b) VQ (c) PCA

Figure 10 Part representation by NMF, VQ, and PCA methods [40]

In the proposed model, we applied NMF to our face data and utilized its basis matrix in the projection part (Figure 8-(b)). NMF finds a low-rank approximation of a matrix V, where $V \approx WH$,

when $V, W,$ and H do not have non-negative values. Given a feature matrix, V, W is a basis matrix that contains basis elements of V, and H is a latent representation matrix. We call the matrix W local weights. These local weights serve as the influence of each vertex on a specific area. We expect that local weights would make the part encodings more semantically meaningful.

McGraw et al. [42] say that enforcing sparsity on the column of H could improve the local separation of features. Thus, we use sparse NMF to express local features more efficiently. We compute this with a sparsity constraint value of 7.5. Figure 11 shows the visualization of the local weights. The bright area shows how much each vertex influences the facial area. Compared to Figure 12, the basis matrix by sparse NMF [46] presents a clear distinction of regions and part-based representation.



(a) Sparsity 0.5 $K = 4$

(b) Sparsity 7.5 $K = 4$

Figure 11 Pre-computed sparse NMF's basis matrix [46]

Figure 12 Pre-computed NMF's basis matrix

Before training the model, we compute a basis matrix W and input a simplified template mesh vertices matrix $V \in \mathbb{R}^{P\times 3}$, where P is the number of vertices with positions in three dimensions, i.e., x, y, and z. This mesh has the last downsampling resolution. Given an input matrix V, NMF produces a basis matrix $W \in \mathbb{R}^{P\times K}$, which means that the basis features of the vertices are indexed by K and the coefficient $H \in \mathbb{R}^{K\times 3}$ is indexed by the vertex positions. Here, $K \in \mathbb{R}$ is the rank, with $K < \min(P, 3)$, i.e., the number of the face part. After obtaining several W, we selected K basis matrices from them that have the most semantic features.

C. Latent Space Manipulation

Projection Matrix Layer

Our encoder takes a whole shape as input and compresses to a low-dimensional representation, i.e., a latent vector. This encoding reflects the whole shape structure. When we factorize the whole encoding, we can generate part encodings corresponding to the shape structure of the part. Thereby, we disentangle different semantic part encodings from the encoding of the whole shapes. We then perform part-level shape manipulation.

Dubrovina et al. [34] use projection matrices to transform a whole shape embedding into semantic part embeddings. They factorize the latent space into a semantic subspace with data-driven

learned parameters. Part-specific projection matrices, $\{P_k\}_{i=1}^{K} \in \mathbb{R}^{N\times N}$, are constrained by a partition of the identity to satisfy two properties: factorization consistency across input data and simple operator of shape composition.

Motivated by this, we use learnable projection matrices to transform the whole part encoding from the global latent space to the localized basis matrix space. Similarly to Dubrovina et al. [34], we define part-specific projection matrices, $\{P_k\}_{i=1}^{K} \in \mathbb{R}^{N\times N}$, where K is the number of semantic parts. Passing through the matrices, the whole part encodings from the encoder are divided into semantic part encodings.

For embedding parts, we implement projection matrices represented as K fully connected layers without biases and with the latent dimension size of $Z \times Z$. The input of the projection layers is a whole face encoding produced by the encoder, and their outputs are K part encodings. The K part encodings can be split unpredictably and have arbitrary meanings. To make them more semantically meaningful, we apply pre-computed local weights. We explain this in the following paragraph.

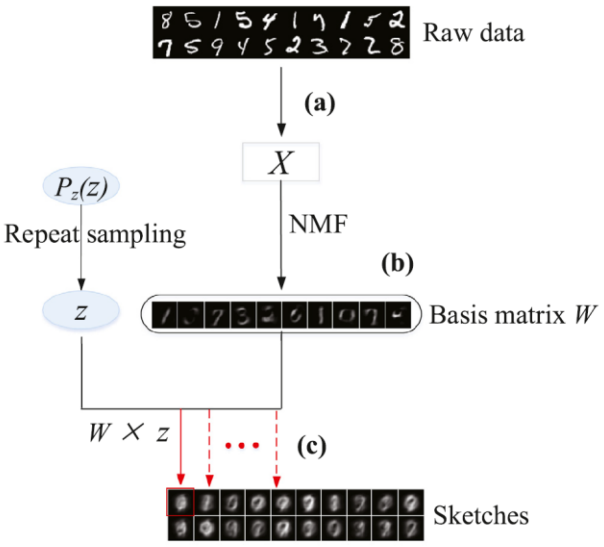


Figure 13 Illustration of constructing sketches [43]

Applying Local Weights to Factorized Part Encodings

Li et al. [43] proposed *sketch*, a combination of random noise and features of the original data, produced by transforming vectors from the noise space to a basis matrix space in NMF. The multiplication of the basis matrix by noise vectors produces transformed new samples consistent with the raw data distribution. As a result, these samples have one or more raw data features (Figure 13).

Inspired by [43], we apply the pre-computed local weights to the part encodings that are factorized by the projection matrices (Figure 9). Each pre-computed local weight is multiplied by each latent vector. Thanks to this operation, each factorized latent vector has a localized weight, and the encodings lie on a part-based subspace. We describe this process schematically in Figure 14.

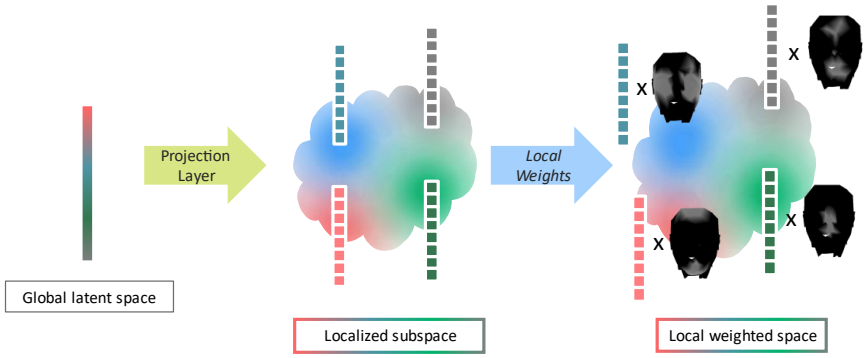


Figure 14 Illustration of the projection part

Pre-computed local weights, which form the basis matrix derived from NMF $W \in \mathbb{R}^{P\times K}$, are applied to the factorized latent vectors $Z \in \mathbb{R}^{K\times Z}$. This process produces a locally weighted matrix $W_L \in \mathbb{R}^{K\times N\times Z}$ and then sum up as $W_L \in \mathbb{R}^{N\times Z}$. This matrix is provided as an input to the fully connected layer of the decoder. Once the first layer of the decoder transforms the input, other processes mirror the encoder with an upsampling procedure, increasing the mesh data approximately four times.

D. Loss Function

To optimize the networks, we exploit the $\mathcal{L}_1$ loss and cycle loss [34]. The $\mathcal{L}_1$ loss is adopted to optimize the autoencoder's reconstruction ability and measure the difference between the predicted mesh and the ground truth data.

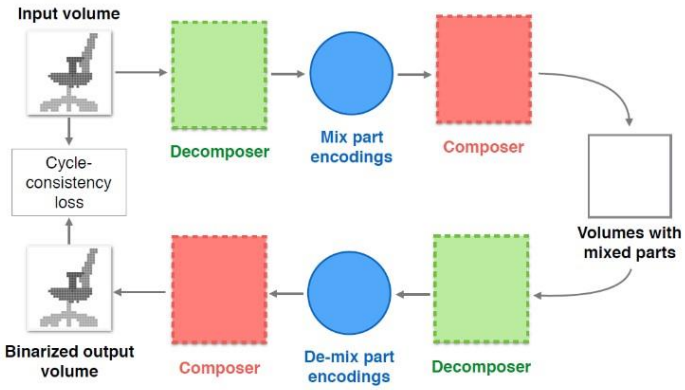


Figure 15 Illustration of the cycle consistency constraint [34]

Cycle loss [34] optimizes the network for the semantically plausible part arrangement using a cycle consistency constraint that encourages $F(G(x)) \approx x$ and $G(F(y)) \approx y$ [47]. Figure 15 illustrates the cycle consistency constraint schematically. In this method, a mini batch of B with training data $\{X\}_{i=1}^{B}$ is encoded by the encoder and factorized to K part encodings. Each part encoding is arbitrarily shuffled in the mini batch, but not mixed with other part encodings. Then, the shuffled mini batch passes through the decoder, which reconstructs them. The reconstruction results are compressed and decompressed by the autoencoder again, but the unmixed part encodings are restored before the second encoder. Finally, the input data and final reconstruction data are compared, and the difference is provided as an error to the backpropagation of the networks.

V. Implementation

A. Datasets

To obtain massive facial mesh data, we used the AFLW2000-3D dataset [48] containing 2,000 3D faces and the corresponding landmarks of AFLW [49] face images (Figure 16). Each 3D face has 53,215 vertices. All faces are in full correspondence and generated by the Basel Face Model [26] without pose variations. In data pre-processing, we matched all facial mesh topology, i.e., those with the same vertex ordering. The dataset was divided into a training set and a test set with 1,780 faces and 220 faces, respectively.

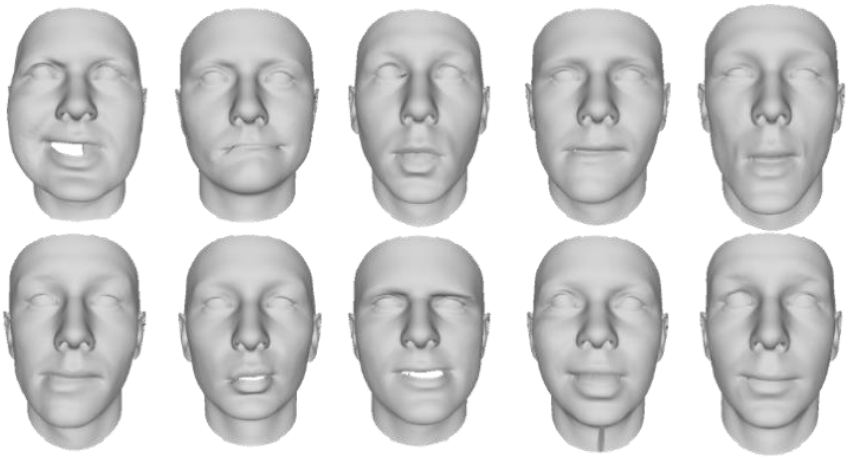


Figure 16 Ten sample faces from the pre-processed AFWL2000-3D dataset [48]

B. Implementation Details

Our proposed model is based on CoMA [23], following their down-sampling and up-sampling method for coarse-to-fine convolution networks. The structure of the encoder and decoder is shown in Table 1. Similar to [23], our encoder contained four convolution layers, followed by a biased ReLU [50]. After passing the convolution layer, the input mesh was down-sampled approximately four times. The last fully-connected layer transformed the face mesh into a 64-dimensional latent vector.

We trained our model for 300 epochs with a batch size of 32. The dimension of the latent vector was 64. The initial learning rate started at 0.0125 and decreased by 0.99 every epoch. We used stochastic gradient descent with a momentum of 0.9 to optimize and set Chebyshev filtering with K as six. We used PyTorch [51] and PyTorch Geometric [52] to implement our model and conducted all experiments with NVIDIA Titan RTX GPU 24GB.

Layer	Input size	Output size
Convolution	53215×3	53125×16
Down-Sampling	53215×16	13304×16
Convolution	13304×16	13304×16
Down-Sampling	13304×16	3326×16
Convolution	3326×16	3326×16
Down-Sampling	3326×16	832×16
Convolution	832×16	832×32
Down-Sampling	832×32	208×32
Fully Connected	208×32	64

Table 1 Encoder architecture

Layer	Input size	Output size
Fully Connected	208×64	208×32
Up-Sampling	832×32	208×32
Convolution	832×16	832×32
Up-Sampling	3326×16	832×16
Convolution	3326×16	3326×16
Up-Sampling	13304×16	3326×16
Convolution	13304×16	13304×16
Up-Sampling	53215×16	13304×16
Convolution	53215×3	53125×16

Table 2 Decoder architecture

VI. Experimental Results

The experimental results of our proposed model are described in this section. We present the practicality of our model with several generation tasks and an ablation study. In all experiments, except for the last ablation study, we set the number of face parts, K, as 4.

A. Generation Results

Part interpolation

In this experiment, we tested the part manipulation results by applying interpolation between source and target (Figure 17). We interpolated the source's part encodings to the target's corresponding part encodings obtained by a factorized latent vector described in chapter IV-C.

Figure 18 shows that as the respective part of the face influence changes, the other parts of the face are not affected. Plus, we expected that each row's changing part matches each local weight in the same row. As a result, we observed that each variation area corresponds to each local weight in Figure 18. Color gradients in the variation area included visualizing the Hausdorff distance between the first face $(\alpha = \frac{1}{9})$ and the last face $(\alpha = \frac{8}{9})$ in each row. Each of them displays a variation of each interpolation more clearly. The blue-colored gradient signifies that the vertices of the source and target are nearby, while the red-colored gradient means they are further away.

However, the variation corresponding to the third local weight (the third row in Figure 18Figure 17) includes the changes in the eyebrow area as well as the nose. We provide two possible explanations for this: the transformation by projection matrices, and the quality of the dataset. First, projection matrices have a role in transforming part encodings from latent space to local weights

space as well as factorizing the latent space. Thus, this transformation would cover unassigned areas by local weights. Second, the variation of eyebrow and nose might be related because this correlation is found across the dataset we used, not specific face data.

(a) Source (b) Target

Figure 17 Source and target face

Figure 18 Result of part manipulation

Reconstruction ability

Figure 19 and Figure 20 show the results of reconstruction using our model and baseline [23], respectively. Overall, our model results are comparable with the baseline's and display the distinctive identity of each ground truth face. We found minor reconstruction errors in the forehead and eyes compared to the baseline results. Nevertheless, our model shows convincing results considering that improving reconstruction ability is not the primary goal in this study. In Figure 19 and Figure 20, color gradients present the Hausdorff distance, which means there is a reconstruction error between ground truths and reconstruction results.

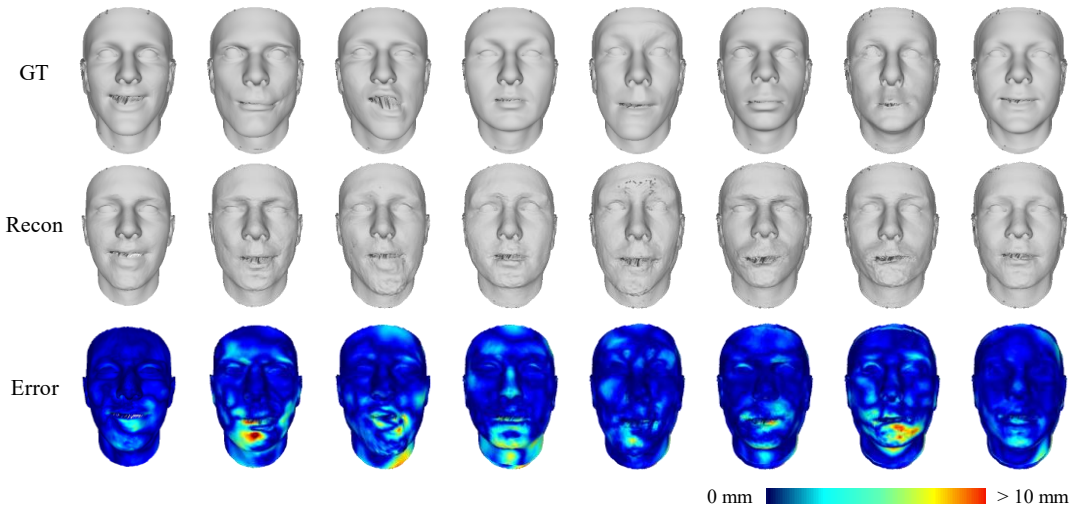


Figure 19 Reconstruction result of our model

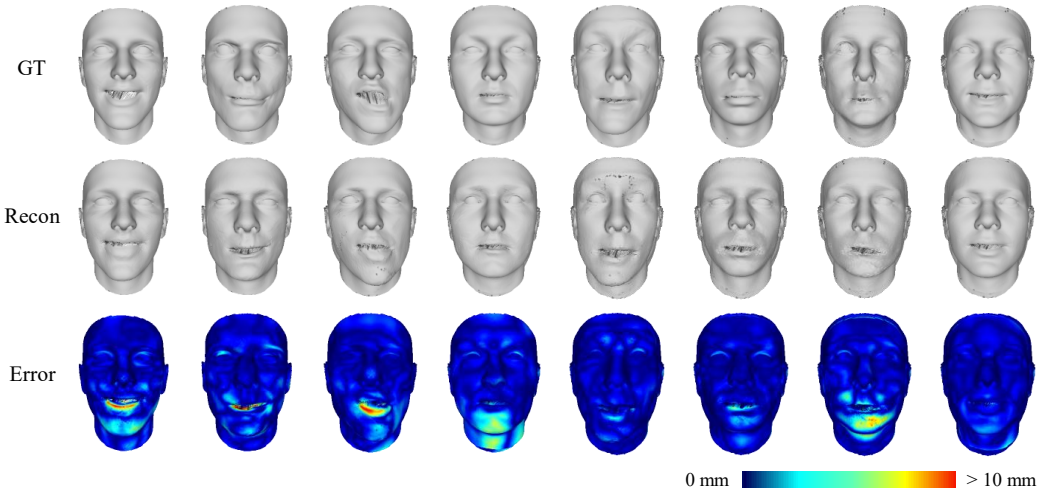


Figure 20 Reconstruction result of baseline [23]

B. Diversity Visualization

One of our goals was to improve the variation of generated data with the proposed method. To demonstrate the variety of data, we measured the diversity of generated data from our model.

Using the trained encoder, we encoded 220 random faces from our training set and test set, respectively. Since our proposed model allows part manipulation and modification, we synthesized 220 faces by combining five source faces and 11 target faces for four parts. The synthesis samples are shown in Figure 21. We assumed that this manipulation would encourage the model to generate more various outputs.

The result was visualized by projecting selected data onto a 2D plane using PCA and t-SNE [53], shown in Figure 22 and Figure 23. We displayed all encoded faces as markers and summarized them with ellipses. Here, there are three types of encoding: training set (red), test set (yellow), and part synthesis (green).

Schor et al. [33] named their test data as unseen data that the model cannot encounter during training. According to them, if the model generates samples like the unseen data, it means that the model can generate diverse data, including what it did not learn. Using this perspective, we focused on the area where our synthesis outputs were placed.

In Figure 22, we can discern that our synthesis sample area (green ellipse) involves both areas of the training set and test set (red and yellow ellipses) in the 2D PCA plane. Figure 23 (t-SNE visualization) presents this result more distinctively as the synthesis samples are also located in a wider region as well as the region of the training set and test set.

In our visualizations, even though the training data and test data overlap, our synthesis samples (green) cover wider areas in the encoding space. As a result, our proposed method shows a prominent performance to extend the model's representation ability.

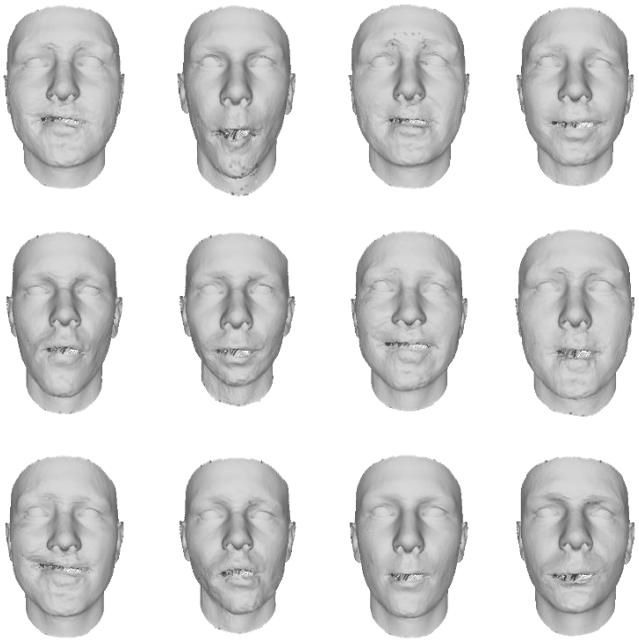


Figure 21 Synthesis face samples

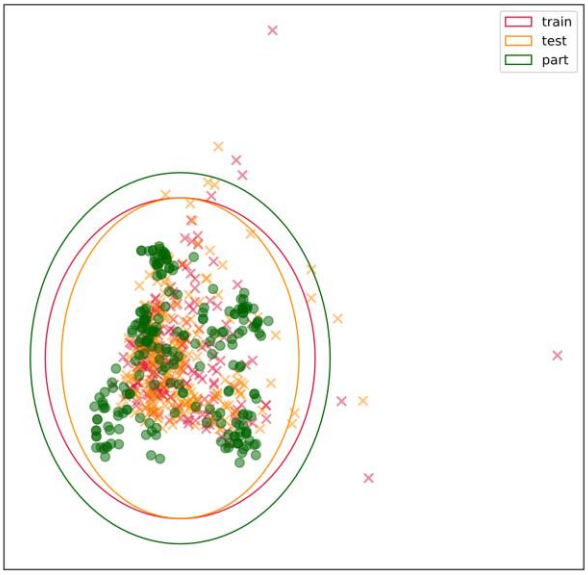


Figure 22 Visualization of 2D PCA plane projection

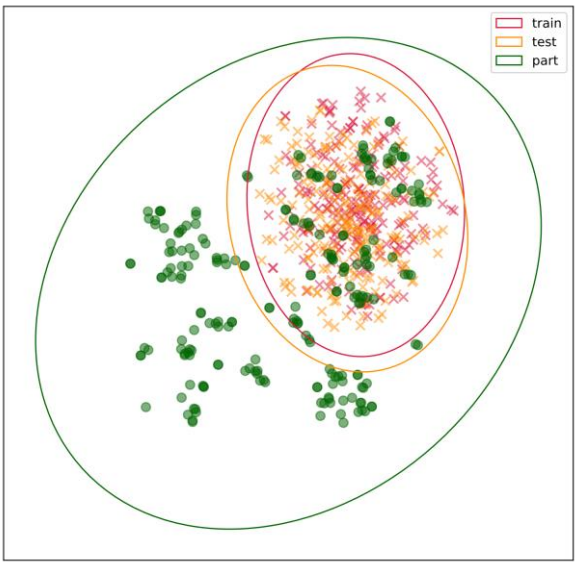


Figure 23 Visualization of 2D t-SNE place projection

C. Ablation Study

To study the effect of each component of our approach, we experimented with an ablation study with several variations of model components, such as local weight, projection matrices, cycle consistency, and the number of projection matrices and local weights.

Without local weights

In the proposed model, local weights were obtained by NMF to make decomposed part spaces more semantically meaningful. To verify the effect of local weights, the model was trained without applying local weights to the projection part. The results are shown in Figure 24 and are presented sequentially. The far-right face visualizes the Hausdorff distance between the first face and the last face in the sequence.

Without applying local weight, the results do not display noticeable changes in some faces, and changing areas of the face also are intertwined with each other and look arbitrary.

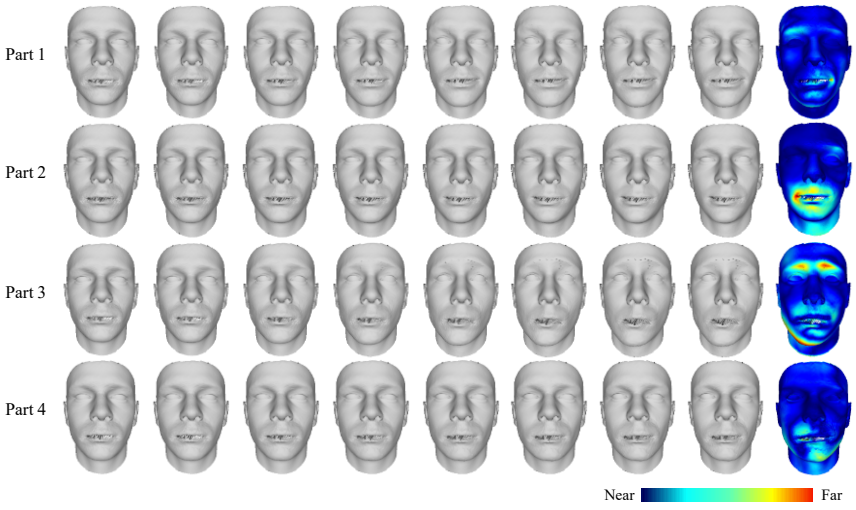


Figure 24 Results of part interpolation without applying local weights

Without projection matrices

In this experiment, we removed the factorization step to explore its impact. There was a major change in the encoder, which is visually described in Figure 25.

In the process, the output $F \in \mathbb{R}^{D_i \times C}$ of the last convolution and the down-sampling layer directly multiplied each pre-computed local weight $W \in \mathbb{R}^{P \times K}$ element by element. D_i is the i-th down-sampling resolution, where i is 4, and C is the last convolution filter size of the encoder. This was then applied to the fully connected layer and transformed into K 64-dimensional latent vectors. Finally, these locally weighted vectors were summed and inputted into the decoder. The result is shown in Figure 26.

Compared to Figure 18, the changing parts of faces in Figure 26 reflect less local weight except for the first face. Specifically, the second face's cheek area impacts the mouth area, and other faces change different parts with local weight. Considering the results, we can infer that the projection matrices help exert local weights better. Accordingly, projection matrices not only factorize the latent space but also transform it into local weights space.

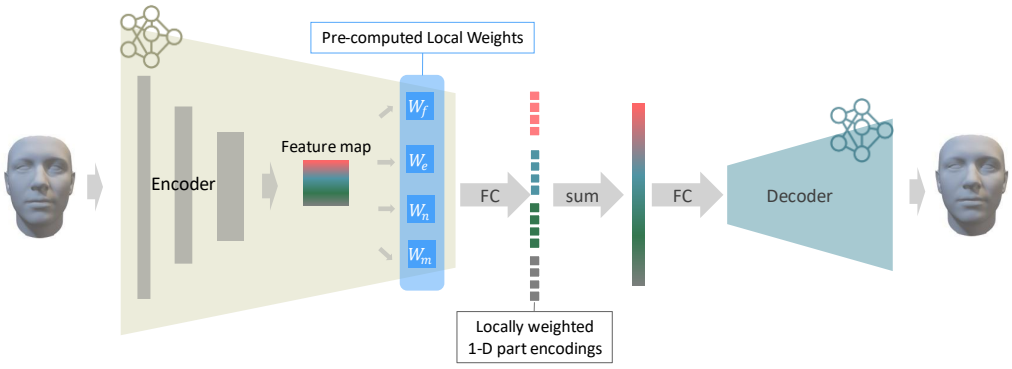


Figure 25 Altered model without projection matrices

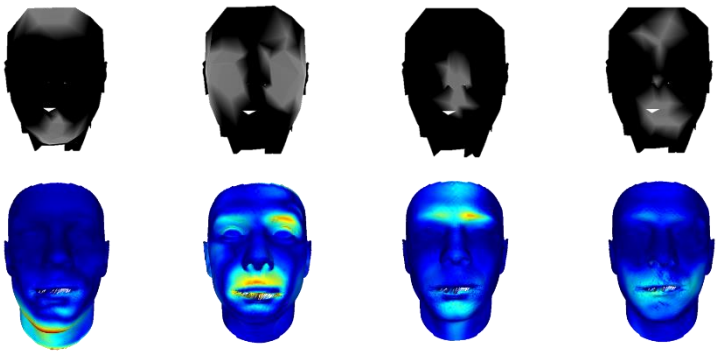


Figure 26 Part interpolation result without factorization by projection matrices

Without cycle consistency constraint

We omitted the cycle consistency constraint [34] during training. Since this constraint expects part separation to be more apparent and plausible, we guessed that part interpolation becomes unclear without the constraint.

The result of the test is shown in Figure 27. It presents a plausible part arrangement in terms of given local weights. However, the variation in the third face is more spread out than the face with cycle consistency constraint. To compare, please see Figure 29.

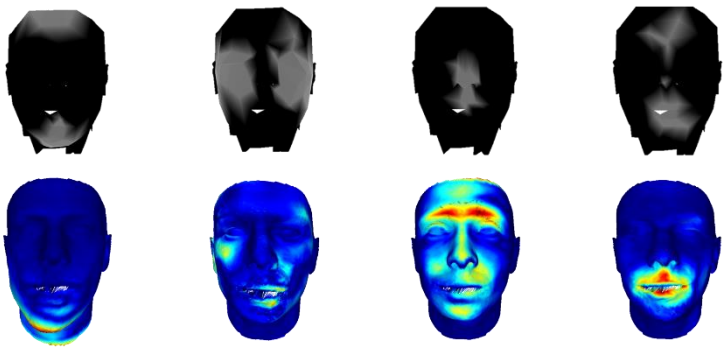


Figure 27 Part interpolation result without cycle consistency constraint

Increasing K: The number of projection matrices and local weights

To prove the influence of the number of local weights, we added one projection matrix and one local weight to divide a face into five parts. In Figure 28, the fifth column is a newly added local weight influencing the neck and side forehead. The generated face with the new local weight clearly shows its changing part is limited to around the neck. Given this result, the proposed local weights from NMF exhibit a local change while generating a holistic shape change in the model.

The number of local weights should be the same or less than the number of projection matrices. Yet, each number of projection matrices and local weights should be matched because of the dimensionality of the produced vectors. If not, the dimensionality does not match the locally weighted factorized vectors.

Figure 29 schematically describes five projection matrices and four local weights. In this condition, the latent vector from the encoder can be factorized into five latent vectors, but only four divided vectors can be multiplied by local weights. This operation changes the multiplied latent vectors' dimensions. Therefore, the four multiplied vectors and the others have different dimensionality. Consequently, unmatched dimensionality prevents subsequent operations.

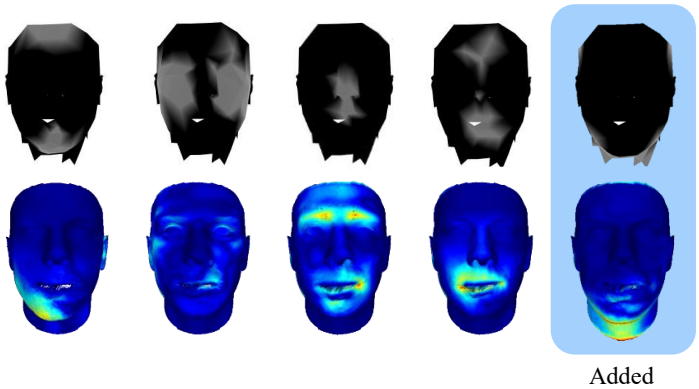


Figure 28 Part interpolation result from applying five projection matrices and local weights

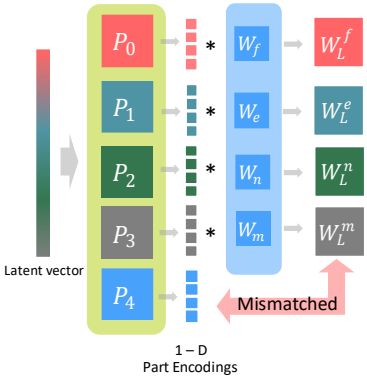


Figure 29 Mismatched dimensionality

D. Discussion

Our proposed model performed notable part manipulation and synthesis using a holistic generative approach. However, there are a few points that need further discussion. The whole model did not show improved reconstruction compared to the baseline [23]. Although this is not critical, there were general noises on the meshes. We assume that used model components, such as projection matrices and local weights can improve the diversity of outputs but might constrain the decoder's improvement.

Concerning the level of model components, the correlation between the changing area of faces and local weights should be better addressed. Most changing areas generally reflect corresponding local weights features, but some include another part or ignores them. One assumption is because of projection matrices that influence factorization and transforms latent vectors. The other is the natural quality of the dataset having correlations between facial features. Although we suggest two possible reasons, these need to be explored thoroughly.

Second, the semantic meaning of local weights needs to be examined. Our local weights were computed algorithmically, not manually segmented or labeled. Therefore, it lacked semantic meaning and detailed segmentation of the human face, such as the separation of eyes and eyebrows.

Finally, we multiply the part encodings in latent space and local weights in NMF. This approach seemed to work in our setting because the projection matrices transform part encoding to local weights' space. We have shown experimentally that our process works, but the mathematical proof is still needed.

VII. Conclusion

In this dissertation, we proposed a locally weighted 3D generative face model using spectral convolution networks for a 3D mesh. We increased the model's expressiveness by manipulating the local part of a face without explicit mesh segmentation. We implemented our generative model based on Ranjan et al.'s [23] autoencoder. We also combined latent space factorization and applied local weights.

We evaluated our proposed model's synthesis ability, reconstruction ability, and diversity visualization. The ablation study also showed each model component's effect on generation results. In detail, our model used latent vector manipulation while applying local weights. This manipulation allowed each part of the face to be modified and the corresponding part between faces that were not identical were interpolated. We observed that the modified areas of the face were separated, and the changes were noticeable. With data encoding visualization, we verified the improvement of the model's representative power through diversity visualization. Our model synthesized samples in the central and peripheral regions of the dataset.

Our model is simpler than existing models that use several part decoders and composition networks. Only a global encoder and decoder were used, and additional networks were not required, meaning that our model performed with a lower number of weight parameters.

In future work, we would like to extend our model to apply other generative models i.e., VAE or GANs, to improve output's quality. Generating face textures with geometry also would express the quality of outcomes better. Besides, it would be worthwhile to study parts-based representation to improve the proposed local weights to develop the model's synthesis ability.

Bibliography

[1] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, “DeepFace: Closing the Gap to Human-Level Performance in Face Verification,” *Proceedings of the IEEE conference on computer vision and pattern recognition(CVPR)*. 2014. p. 1701-1708.

[2] A. Chen, Z. Chen, G. Zhang, K. Mitchell, and J. Yu, “Photo-Realistic Facial Details Synthesis From Single Image,” in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South), Oct. 2019, pp. 9428–9438

[3] B. Gecer *et al.*, “Synthesizing Coupled 3D Face Modalities by Trunk-Branch Generative Adversarial Networks,” in *Proceedings of European Conference on Computer Vision*, Sep. 2020, pp. 415--433

[4] J. Kim, M. G. Choi, and Y. J. Kim, “Real-time Muscle-based Facial Animation using Shell Elements and Force Decomposition,” in *Symposium on Interactive 3D Graphics and Games*, San Francisco CA USA, May 2020, pp. 1–9

[5] L. Hu *et al.*, “Avatar digitization from a single image for real-time rendering,” *ACM Trans. Graph.*, vol. 36, no. 6, pp. 1–14, Nov. 2017

[6] G. Bouritsas, S. Bokhnyak, S. Ploumpis, M. Bronstein, and S. Zafeiriou, “Neural 3D Morphable Models: Spiral Convolutional Networks for 3D Shape Representation Learning and Generation,” *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 2019. p. 7213-7222.

[7] A. S. Jackson, A. Bulat, V. Argyriou, and G. Tzimiropoulos, “Large Pose 3D Face Reconstruction from a Single Image via Direct Volumetric CNN Regression,” in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, Oct. 2017, pp. 1031–1039

[8] S. Zhang, H. Yu, J. Dong, T. Wang, Z. Ju, and H. Liu, “Automatic Reconstruction of Dense 3D Face Point Cloud with a Single Depth Image,” in *2015 IEEE International Conference on Systems, Man, and Cybernetics*, Oct. 2015, pp. 1439–1444

[9] R. Li et al., “Learning Formation of Physically-Based Face Attributes,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*. 2020. p. 3410-3419.

[10] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, “Geometric deep learning: going beyond Euclidean data,” *IEEE Signal Process. Mag.*, vol. 34, no. 4, pp. 18–42, Jul. 2017

[11] M. Defferrard, X. Bresson, and P. Vandergheynst, “Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering,” *Proceedings of the 30th International Conference on Neural Information Processing Systems.* 2016. p. 3844-3852.

[12] F. Monti, D. Boscaini, J. Masci, E. Rodola, J. Svoboda, and M. M. Bronstein, “Geometric Deep Learning on Graphs and Manifolds Using Mixture Model CNNs,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, Jul. 2017, pp. 5425–5434

[13] R. Hanocka, A. Hertz, N. Fish, R. Giryes, S. Fleishman, and D. Cohen-Or, “MeshCNN: A Network with an Edge,” *ACM Trans. Graph.*, vol. 38, no. 4, pp. 1–12, Jul. 2019

[14] J. Schult, F. Engelmann, T. Kontogianni, and B. Leibe, “DualConvMesh-Net: Joint Geodesic and Euclidean Convolutions on 3D Meshes,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*. 2020. p. 8612-8622.

[15] J. Wu, C. Zhang, T. Xue, B. Freeman, and J. Tenenbaum, “Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling,” *Advances in neural information processing systems*, 2016, 29: 82-90.

[16] H. Wang, N. Schor, R. Hu, H. Huang, D. Cohen-Or, and H. Huang, “Global-to-local generative model for 3D shapes,” *ACM Trans. Graph.*, vol. 37, no. 6, pp. 1–10, Jan. 2019

[17] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. Guibas, “Learning Representations and Generative Models for 3D Point Clouds,” *International conference on machine learning*. PMLR, 2018. p. 40-49.

[18] 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*, 1999, pp. 187–194

[19] J. R. Tena, F. De la Torre, and I. Matthews, “Interactive region-based linear 3D face models,” in *ACM SIGGRAPH 2011 papers on - SIGGRAPH ’11*, 2011, p. 1

[20] L. Tran, F. Liu, and X. Liu, “Towards High-Fidelity Nonlinear 3D Face Morphable Model,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019, pp. 1126–1135

[21] C.-H. Lee, Z. Liu, L. Wu, and P. Luo, “MaskGAN: Towards Diverse and Interactive Facial Image Manipulation,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020, pp. 5548–5557

[22] K. Khan, M. Mauro, P. Migliorati, and R. Leonardi, “Head pose estimation through multi-class face segmentation,” in *2017 IEEE International Conference on Multimedia and Expo (ICME)*, Jul. 2017, pp. 175–180

[23] A. Ranjan, T. Bolkart, S. Sanyal, and M. J. Black, “Generating 3D Faces Using Convolutional Mesh Autoencoders,” in *Proceedings of the European Conference on Computer Vision (ECCV).* 2018. p. 704-720.

[24] B. Egger *et al.*, “3D Morphable Face Models -- Past, Present and Future,” *ArXiv190901815 Cs*, Sep. 2019

[25] J. Booth, A. Roussos, S. Zafeiriou, A. Ponniah, and D. Dunaway, “A 3D Morphable Model Learnt from 10,000 Faces,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 5543–5552

[26] 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

[27] V. F. Abrevaya, S. Wuhrer, and E. Boyer, “Multilinear Autoencoder for 3D Face Model Learning,” in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Mar. 2018, pp. 1–9

[28] V. F. Abrevaya, A. Boukhayma, S. Wuhrer, and E. Boyer, “A Decoupled 3D Facial Shape Model by Adversarial Training,” in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South), Oct. 2019, pp. 9418–9427

[29] S. Moschoglou, S. Ploumpis, M. Nicolaou, A. Papaioannou, and S. Zafeiriou, “3DFaceGAN: Adversarial Nets for 3D Face Representation, Generation, and Translation,” *ArXiv190500307 Cs*, May 2019

[30] S. Cheng, M. Bronstein, Y. Zhou, I. Kotsia, M. Pantic, and S. Zafeiriou, “MeshGAN: Non-linear 3D Morphable Models of Faces,” *ArXiv190310384 Cs*, Mar. 2019

[31] K. Li, J. Liu, Y. Lai, and J. Yang, “Generating 3D Faces using Multi-column Graph Convolutional Networks,” *Comput. Graph. Forum*, vol. 38, no. 7, pp. 215–224, Oct. 2019

[32] D. Ghafourzadeh, C. Rahgoshay, A. Beauchamp, A. Aubame, and T. Popa, “Part-Based 3D Face Morphable Model with Anthropometric Local Control,” 2019

[33] N. Schor, O. Katzir, H. Zhang, and D. Cohen-Or, “CompoNet: Learning to Generate the Unseen by Part Synthesis and Composition,” in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2019, pp. 8758–8767

[34] A. Dubrovina, F. Xia, P. Achlioptas, M. Shalah, R. Groscot, and L. Guibas, “Composite Shape Modeling via Latent Space Factorization,” *Proceedings of the IEEE International Conference on Computer Vision.* 2019. p. 8140-8149.

[35] M. Jaderberg, K. Simonyan, and A. Zisserman, “Spatial Transformer Networks,” *Advances in neural information processing systems*. 2015. p. 2017-2025.

[36] C. Öngün and A. Temizel, “LPMNet: Latent Part Modification and Generation for 3D Point Clouds,” *ArXiv200803560 Cs*, Aug. 2020

[37] E. Collins, R. Bala, B. Price, and S. Susstrunk, “Editing in Style: Uncovering the Local Semantics of GANs,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020, pp. 5770–5779

[38] K. Hornik, I. Feinerer, M. Kober, and C. Buchta, “Spherical k-Means Clustering,” *Journal of Statistical Software*, 2012, 50.10: 1-22.

[39] E. Collins and S. Süsstrunk, “Deep Feature Factorization for Content-Based Image Retrieval and Localization,” in *2019 IEEE International Conference on Image Processing (ICIP)*, Sep. 2019, pp. 874–878

[40] D. D. Lee and H. S. Seung, “Learning the parts of objects by non-negative matrix factorization,” *Nature*, vol. 401, no. 6755, pp. 788–791, Oct. 1999

[41] W. P. Koppen, W. J. Christmas, D. J. M. Crouch, W. F. Bodmer, and J. V. Kittler, “Extending non-negative matrix factorisation to 3D registered data,” in *2016 International Conference on Biometrics (ICB)*, Jun. 2016, pp. 1–8

[42] T. McGraw, J. Kang, and D. Herring, “Sparse Non-Negative Matrix Factorization for Mesh Segmentation,” *International Journal of Image and Graphics.*, vol. 16, no. 01, p. 1650004, Jan. 2016

[43] W. Li *et al.*, “Sketch-then-Edit Generative Adversarial Network,” *Knowledge-Based Systems*, 2020, 106102.

[44] D. K. Hammond, P. Vandergheynst, and R. Gribonval, “Wavelets on graphs via spectral graph theory,” *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 129–150, Mar. 2011

[46] W. Xu, X. Liu, and Y. Gong, “Document Clustering Based On Non-negative Matrix Factorization,” *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval.* 2003. p. 267-273.

[47] V. K. Potluru, S. M. Plis, J. L. Roux, B. A. Pearlmutter, V. D. Calhoun, and T. P. Hayes, “Block Coordinate Descent for Sparse NMF,” *ArXiv13013527 Cs*, Mar. 2013

[47] V. K. Potluru, S. M. Plis, J. L. Roux, B. A. Pearlmutter, V. D. Calhoun, and T. P. Hayes, “Block Coordinate Descent for Sparse NMF,” *Proceedings of the IEEE international conference on computer vision(ICCV)*. 2017. p. 2223-2232.

[49] X. Zhu, Z. Lei, X. Liu, H. Shi, and S. Z. Li, “Face Alignment Across Large Poses: A 3D Solution,” *Proceedings of the IEEE conference on computer vision and pattern recognition(CVPR).* 2016. p. 146-155.

[50] M. Köstinger, P. Wohlhart, P. M. Roth, and H. Bischof, “Annotated Facial Landmarks in the Wild: A large-scale, real-world database for facial landmark localization,” in *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, Nov. 2011, pp. 2144–2151

[51] X. Glorot, A. Bordes, and Y. Bengio, “Deep Sparse Rectifier Neural Networks,” *Proceedings of the fourteenth international conference on artificial intelligence and statistics.* 2011. p. 315-323.

[52] A. Paszke *et al.*, “PyTorch: An Imperative Style, High-Performance Deep Learning Library,” *Advances in neural information processing systems.* 2019. p. 8026-8037.

[53] M. Fey and J. E. Lenssen, “Fast Graph Representation Learning with PyTorch Geometric,” *ArXiv190302428 Cs Stat*, Apr. 2019

[54] L. van der Maaten and G. Hinton, “Visualizing Data using t-SNE,” *Journal of machine learning research*, 2008, 9.Nov: 2579-2605.

국문 초록

김민영

인공지능 · 소프트웨어학부

이화여자대학교 대학원

사람의 얼굴을 3 차원에서 표현하고 다양한 얼굴상을 만들어내려는 얼굴 생성 모델 연구가 진행되어왔다. 특히 적대적 생성 신경망 (Generative Adversarial Networks)의 등장으로 생성 모델에 대한 관심은 더욱 높아지고 있으며 딥러닝을 활용한 연구도 활발히 진행되고 있다. 얼굴 생성은 가상 아바타 및 가상 인물 콘텐츠 생성에 활용될 수 있다.

얼굴 생성에 관련한 기존 연구들은 얼굴 전체를 활용하여 다양한 얼굴을 만들려는 시도이다. 그러나 얼굴을 의미론적 특징을 가진 여러 개의 부분으로 나눠 생성하는 것이 모델의 표현 능력을 보다 향상시키고 부족한 얼굴 데이터 셋의 한계를 해결할 수 있다.

본 학위 논문에서는 국소적 가중치를 적용한 3 차원 얼굴 생성 모델을 제안한다. 전체 얼굴 메쉬를 학습하여 부분적 조작이 불가능한 기존 연구들과는 달리 제안하는 모델은 전체 얼굴 메쉬를 학습하면서도 얼굴의 부분적 조작이 가능하다. 이는 데이터가 학습된 기저 공간을 여러 개의 부분 공간으로 분해함으로써 부분 공간에 해당하는 기저 벡터를 선형적으로 변형함으로써 가능하다. 또한, 비음 행렬 분해 (Non-negative matrix factorization) 알고리즘으로 생성한 국소적 가중치를 기저 공간에 적용하여 분해된 부분 공간이 좀 더 의미 있는 부분적 얼굴 표현을 가질 수 있도록 한다.

제안하는 모델을 구현하고 실험하여 효과적인 얼굴 부분 조작이 가능함을 확인하였으며 모델의 생성 능력 또한 향상하였음을 보였다. 또한, 비교 실험을 통해

본 연구에서 제안한 국소적 가중치를 적용한 기저 공간 분해가 의미 있는 결과를 생성함을 확인할 수 있었다.

감사의 글

많은 분들의 도움이 있었기에 무사히 석사 과정을 마칠 수 있었습니다. 이 글을 통해 감사의 인사를 드리고자 합니다.

가장 먼저 지도해주신 김영준 교수님께 감사의 말씀 드립니다. 부족한 점이 많은 저에게 기회를 주시고 믿고 큰 가르침 주심에 감사드립니다. 교수님의 지도를 통해 연구란 무엇인지와 이에 임하는 자세를 배우고, 또 채울 점을 깨닫고 발전하려 노력할 수 있었습니다. 깊은 가르침 주신 것 잊지 않고 앞으로도 더욱 노력하여 보시기에 부끄럽지 않은 연구자가 되겠습니다. 또한 바쁘신 가운데 학위 논문의 심사위원을 맡아주시고 2 년 동안 수업으로, 연구 코멘트로 큰 도움 주신 민동보, 오유란 교수님께도 깊은 감사의 말씀 드립니다.

2 년 동안 가족이 되어 주었던 GLAB 여러분 감사드립니다. 부족한 저를 아낌없이 도와주셨던 김예솔 언니, 민혜정 언니, 송다은, 김시연, 한경민 박사님, 김관식 박사님, 유지연 선생님, 그리고 졸업하신 김여진 언니, 김지수 언니, 김정민, 김지선 모두 감사드립니다. 또한 석사 초기 여러모로 신경써주신 김은희 선생님 감사드립니다.

즐거울 때나 힘들 때나 응원해 주는 김주영, 강보민, 이현정, 정다운, 미국에 있는 김예경, Stephanie, 그리고 지면에 적지 못한 모든 친구들에게 감사를 전합니다. 항상 고민많고 우유부단한 제 이야기를 들어주고 지지해주어 감사합니다.

언제나 저의 든든한 버팀목인 부모님께 늘 사랑하고 감사하다는 말 드립니다. 멀리서 공부한다고 늘 신경써주시는 외할머니, 이모들, 사촌들, 사랑하는 가족들에게 감사드립니다. 서울 살이에 언제나 쉴 곳이 되어주었던 작은 오빠, 그리고 혜지 언니에게 감사드립니다.

지면에는 언급하지 못했지만, 저를 지지하고 격려 해주신 모든 분들께도 진심으로 감사말씀 드립니다. 앞으로도 더욱 성장하여 제가 받은 도움을 또 다른 이에게 나누는 사람이 되겠습니다.

김민영 올림