Disentangled Representation Learning for 3D Face Shape

Zi-Hang Jiang, Qianyi Wu, Keyu Chen, Juyong Zhang*
University of Science and Technology of China
{jzh0103, wqy9619, cky95}@mail.ustc.edu.cn juyong@ustc.edu.cn

Abstract

In this paper, we present a novel strategy to design disentangled 3D face shape representation. Specifically, a given 3D face shape is decomposed into identity part and expression part, which are both encoded in a nonlinear way. To solve this problem, we propose an attribute decomposition framework for 3D face mesh. To better represent face shapes which are usually nonlinear deformed between each other, the face shapes are represented by a vertex based deformation representation rather than Euclidean coordinates. The experimental results demonstrate that our method has better performance than existing methods on decomposing the identity and expression parts. Moreover, more natural expression transfer results can be achieved with our method than existing methods.

1. Introduction

A 3D face model is comprised of several components like identity, expression, appearance, pose, *etc*., and the 3D face shape is determined by identity and expression attributes [20]. Decoupling 3D face shape into these two components is an important problem in computer vision as it could benefit many applications like face component transfer [42, 36], face animation [13, 35], avatar animation [22], *etc*. The aim of this paper is to develop an attribute decomposition model for 3D face shape such that a given face shape can be well represented by its identity and expression part.

Some existing 3D face parametric models already represent face shapes by the identity and expression parameters. Blanz and Vetter proposed 3D Morphable Model (3DMM) [5] to model face shapes. The most popular form of 3DMM is a *linear* combination of identity and expression basis [2, 43]. FaceWareHouse [14] adopts the *bilinear* model and constructs face shapes from a tensor with identity and expression weights. Recently, FLAME [26] utilizes articulated model along attributes like the jaw, neck *et al*. to achieve the state-of-the-art result. A common characteristic of these linear and bilinear models is that each attribute lies in individual *linear* space and their combination from each attribute is also *linear*. Linear statistical models have limitations like limited expression ability and disentanglement. This limitation comes from the linear formulation itself [38, 3]. However, facial variations are nonlinear in the real world, *e.g*., the variations in different facial expressions. Although some recent works [8, 7, 6, 28, 24] are proposed to improve statistical models, they still construct the 3D face shape by linearly combining the basis.

Inspired by rapid advances of deep learning techniques, learning-based approaches have been proposed to embed 3D face shape into nonlinear parameter spaces, and the representation ability of these methods gets greatly improved, *e.g*., being able to represent geometry details [4], or reconstructing whole face shapes using very few parameters [3]. However, all of these methods encode the entire face shape into one vector in the latent space, and thus cannot distinguish the identity and expression separately. On the other hand, many applications like animation [12], face retargeting [37, 35], and more challenging task like 3D face recognition [31, 27] need to decompose 3D face shape into identity and expression component.

In this paper, we aim to build a disentangled parametric space for 3D face shape with powerful representation ability. Some classical linear methods [5, 14] have already decomposed expression and identity attributes, while they are limited by the representation ability of linear models. Although deep learning based method is regarded as a potential enhancement way, how to design the learning method is not straightforward *e.g*. the neural network structure and the 3D face shape representation features for deep learning. Besides, another challenging issue is that how to make use of the identity and expression labels in the existing datasets like FaceWareHouse [14] for the network training.

To restate the problem, assuming that the identity and expression are separately encoded as vector z_{id} and z_{exp}, the linear model like 3DMM decodes the shape via a linear transformation in the form $\bar{S} + A_{id}z_{id} + A_{exp}z_{exp}$, where $\bar{S}$ is mean shape, A_{id} and A_{exp} are the identity and expression

[*] corresponding author

PCA basis. Considering its non-linear nature, we propose to recover the shape via a nonlinear decoder in the form $F(D_{id}(z_{id}), D_{exp}(z_{exp}))$, where $D_{id}(\cdot), D_{exp}(\cdot)$ and $F(\cdot)$ are nonlinear mapping functions learned by the deep neural network. For this learning task, we develop a general framework based on *spectral graph convolution* [18], which allows inputting vertex based feature on the mesh and decouples 3D face shape into separated attribute components. Considering that different face shapes are mainly caused by deformations, we propose to represent the input face shape of the neural network with vertex based deformation rather than Euclidean coordinates. The vertex based deformation representation for 3D shape is proposed in [21, 34, 41], which captures local deformation gradient and is defined on vertices. In our experiments, vertex based deformation representation can greatly improve the representation ability, and make the shape deformation more natural. In summary, the main contributions of this paper include the following aspects:

- We propose to learn a disentangled latent space for 3D face shape that enables semantic edit in identity and expression domains.
- We propose a novel framework for the disentangling task defined on 3D face mesh. Vertex-based deformation representation is adopted in our framework, and it achieves better performance than Euclidean coordinates.
- Experimental results demonstrate that our method can achieve much better results in disentangling identity and expression. Therefore, applications like expression transfer based on our method can get more satisfying results.

2. Related Work

Linear 3D Face Shape Models Since the similar work of 3DMM [5], linear parametric models are widely used to represent the 3D face shapes. Vlasic *et al.* [40] propose a multi-linear model to decouple attributes into different modes and Cao *et al.* [14] adopt a bilinear model to represent 3D face shape via identity and expression parameters. Recently, other methods were proposed for further improvement. E.g, by using a large scale dataset to improve 3DMM ability [6], or using an articulated model to better capture middle-end of face [26].

Nonlinear 3D Face Models Recently, some works propose to embed the 3D face shapes by the nonlinear parametric model with the powerfulness of deep learning based method. Liu *et al.* [27] propose a multilayer perceptron to learn a residual model for 3D face shape. Tran [38] put forward an encoder-decoder structure for 3D face shape, which is a part of the nonlinear form of 3DMM. Bagautdinov *et al.* [4] propose a compositional Variational Autoencoder structure for representing geometry details in different levels. Tewari *et al.* [4] generate 3D face by self-supervised approach. Anurag *et al.* [3] propose a graph-based convolutional autoencoder for 3D face shape. These works adopt deep neural network to learn a new parametric latent space for 3D face shape, while none of them consider the problem of face attribute decoupling.

Deep Learning for 3D Shapes Analysis Deep learning based method for 3D shapes analysis attracts more and more attentions in recent years [10]. Masci *et al.* [29] first propose mesh convolutional operations for local patches in geodesic polar coordinates. Sinha *et al.* [33] use geometry image to represent Euclidean parametrization of a 3D object. Monti and Boscaini *et al.* [30] introduce d-dimensional pseudo-coordinates that define a local system around each point with weight functions in the spatial domain. Tan *et al.* [34] apply spatial graph convolution to extract localized deformation components of mesh. Bruna *et al.* [11] first propose spectral graph convolution by exploiting the connection between graph Laplacian and the Fourier basis. Defferrard *et al.* [18] further improve the computation speed of spectral graph convolution by truncated Chebyshev polynomials. In our framework, we adapt fast spectral graph convolution operator for shape attribute extraction. To the best of our knowledge, this is the first deep learning based method for the disentangling task defined on 3D mesh data.

3. Disentangled 3D Face Representation

3.1. Overview

Given a collection of 3D face meshes, we aim to obtain a compact representation of identity and expression. A common observation in expression analysis [15] is that human expressions lie in a high-dimension manifold, and an

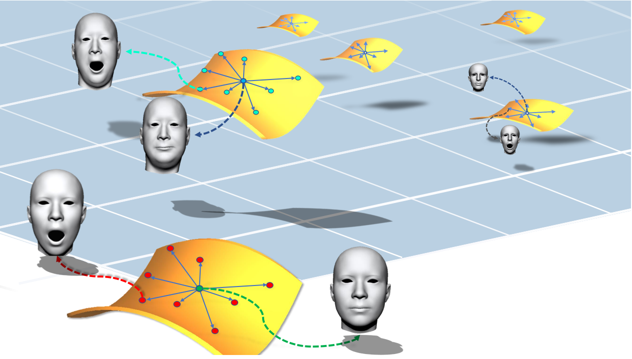


Figure 1. 3D face shape space illustration. As observed in [15], the human expression should lie in a manifold. Based on that, we illustrate each 3D face lie in its expression manifold. Those expression manifolds of different identities should be similar [15, 19].

illustration is shown in Fig. 1 where expression manifold of each individual is rendered in yellow. As the expression manifolds of different individuals are similar [19], an expression of one person could be translated to the same expression on the *mean* face. On the other hand, each individual has its *neutral* expression, which is set as the origin point in each manifold and used to represent his/her identity attribute. Likewise, the same expression on *mean* face represents her/his expression attribute. These two meshes are denoted as *identity mesh* and *expression mesh* respectively.

Based on this observation, our disentangled 3D face representation includes two parts: decomposition and fusion networks. Decomposition network disentangles attributes by decoupling the input face mesh into identity mesh and expression mesh. And the fusion network recovers the original face mesh from identity mesh and expression mesh.

We define a facial mesh as graph structure with a set of vertices $\mathcal{V}$ and edges, $\mathcal{M} = (\mathcal{V}, A)$ with $|\mathcal{V}| = n$. $A \in \{0,1\}^{n\times n}$ represents the adjacency matrix, where $A_{ij} = 1$ denotes an edge connection between vertex v_i and v_j, and $A_{ij} = 0$ otherwise. In our framework, the facial meshes in the training data set contain the same connectivity, and each vertex is associated with a feature vector $\mathbb{R}^d$. The graph feature of mesh $\mathcal{M}$ is denoted as $\mathcal{G} \in \mathbb{R}^{|\mathcal{V}|\times d}$. In our proposed method, a 3D face mesh $\mathcal{M}$ is paired with two meshes, *identity mesh* $\mathcal{M}_{id}$ and *expression mesh* $\mathcal{M}_{exp}$. The triplet $(\mathcal{M}, \mathcal{M}_{id}, \mathcal{M}_{exp})$ will be used for training our networks.

Spectral Graph Convolution Like convolution (correlation) operator for regular 2D image, we adopt a graph convolution operator, *spectral graph convolution*, for extracting useful vertex feature on mesh. We first provide some background about this convolution, and more details can be found in [11, 18, 23].

As we define our mesh $\mathcal{M} = (\mathcal{V}, A)$ in graph structure, the *normalized Laplacian* matrix can be defined as $L = I - D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$, where D is the degree matrix, specifically, a diagonal matrix with $D_{i,i} = \sum_{j=1}^{n} A_{i,j}$ and I stands for identity matrix. Spectral graph convolution defined on graph Fourier transform domain, which is eigenvectors U of laplacian matrix L: $L = U\Lambda U^T$. The convolution on Fourier space is defined as $x * y = U((U^T x) \otimes (U^T y))$, where $\otimes$ is the element-wise Hadamard product. It follows that a signal x is filter by g_θ as $y = g_\theta(L)x$. An efficient way in computation of spectral convolution is parametrized g_θ as a Chebyshev polynomial of order K, like input $x \in \mathbb{R}^{n\times F_{in}}$:

$$y_j = \sum_{i=1}^{F_{in}} \sum_{k=0}^{K-1} \theta_{i,j}^k T_k(\tilde{L}) x_i, \tag{1}$$

where y_j is the j-th feature of $y \in \mathbb{R}^{n\times F_{out}}$, $\tilde{L} = 2L/\lambda_{max} - I_n$ is a scaled Laplacian matrix, λ_{max} is the maximum eigenvalue, T_k is the Chebyshev polynomial of order k and can be compute recursively as $T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$ with $T_0 = 1$ and $T_1 = x$. Each convolution layer has $F_{in} \times F_{out}$ vector of Chebyshev coefficients, $\theta_{i,j} \in \mathbb{R}^k$, as trainable parameters.

Deformation Representation In existing 3D face shape representation works [5, 14, 26, 3], Euclidean coordinate in $\mathbb{R}^3$ is the most common used vertex feature. With spectral graph convolution, we can use other features defined on the vertex. As pointed out in [25], spectral graph convolution is a special form of Laplacian smoothing. Since the main difference among different facial meshes is mainly caused by non-rigid deformations, we prefer a vertex feature related to local deformation rather than the widely used Euclidean coordinate. In this work, we adopt a recent *deformation representation* (DR) [21, 41] to model 3D mesh. We choose neutral expression of mean face as reference mesh, and others are treated as deformed meshes. We briefly introduce the details on how to compute DR feature for a given deformed mesh.

Let us denote the position of the i^{th} vertex v_i on the reference mesh as $\mathbf{p}_i$, and the position of v_i on the deformed mesh as $\mathbf{p}'_i$. The deformation gradient in the 1-ring neighborhood of v_i from the reference model to the deformed model is defined as the affine transformation matrix $\mathbf{T}_i$ that minimizes the following energy:

$$E(\mathbf{T}_i) = \sum_{j\in\mathcal{N}_i} c_{ij} \|(\mathbf{p}'_i - \mathbf{p}'_j) - \mathbf{T}_i(\mathbf{p}_i - \mathbf{p}_j)\|^2 \tag{2}$$

where $\mathcal{N}_i$ is the 1-ring neighborhood of vertex v_i and c_{ij} is the cotangent weight depending only on the reference model to cope with irregular tessellation [9]. By polar decomposition $\mathbf{T}_i = \mathbf{R}_i\mathbf{S}_i$, $\mathbf{T}_i$ can be decomposed into a rotation part $\mathbf{R}_i$ and a scaling/shear part $\mathbf{S}_i$, where rotation can be represent as rotating around the axis ω_i by angle θ_i. We collect non-trivial entries in the rotation and scale/shear components, and obtain the deformation representation of i^{th} vertex in deformed mesh as a $\mathbb{R}^9$ vector. The DR feature of a mesh can treat as a graph feature $\mathcal{G} \in \mathbb{R}^{|\mathcal{V}|\times 9}$ when $d = 9$.

3.2. Decomposition Networks

The input of decomposition networks is deformation representation feature $\mathcal{G}$ of 3D face mesh, and our goal is to disentangle it into identity and expression attributes. It is equivalent to map the input mesh $\mathcal{M}$ to the other two triplet elements $(\mathcal{M}_{id}, \mathcal{M}_{exp})$.

Decomposition part includes two parallel networks with the same structure, one for extracting expression mesh $\mathcal{M}_{exp}$ and the other for extracting identity mesh $\mathcal{M}_{id}$. Taking the identity branch as an example, the input will go through several spectral graph convolution layers for mesh

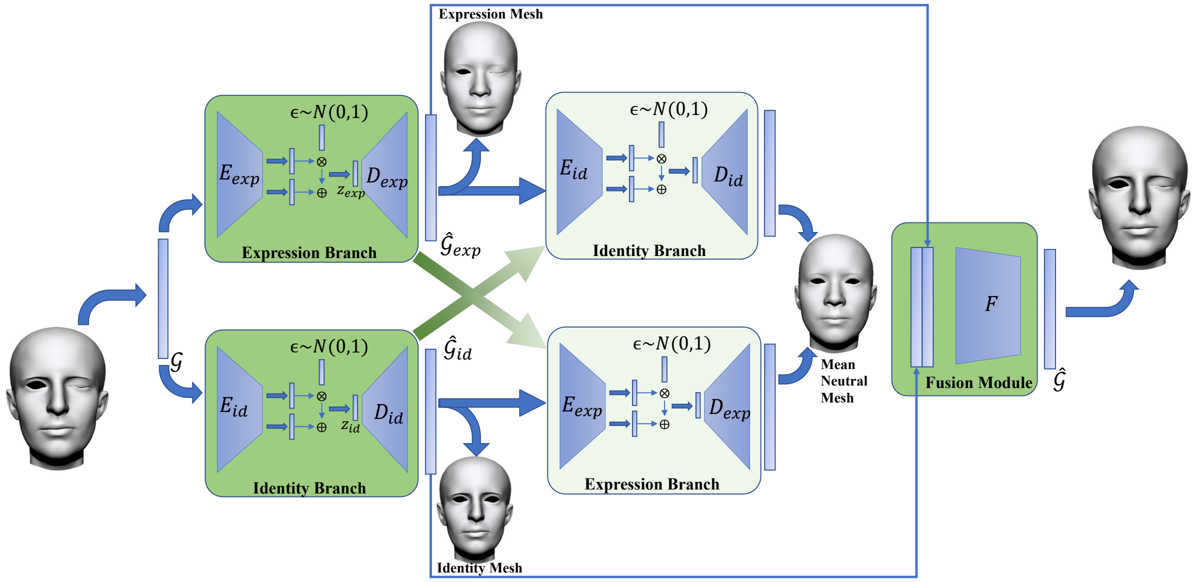


Figure 2. Framework overview. Our network includes two parts, the decomposition part and the fusion part. There are two branches in the decomposition part, one for expression extraction and the other one for identity extraction. Fusion module targets for recovering original mesh from the output of the decomposition part.

feature extraction, with a bottleneck architecture of fully connected layers as an encoder-decoder structure. This structure is applied to obtain latent identity representation.

The output should be close to DR feature of $\mathcal{M}_{id}$. The same structure and principle are applied on expression branch to obtain expression mesh $\mathcal{M}_{exp}$. We use the bottleneck layer in encoder-decoder part for each branch as a new compact parametric space for the corresponding attribute. These two branches accomplish attribute disentanglement task as shown in Fig.2.

We denote $\mathcal{G}_{id}$ as the deformation representation of identity mesh $\mathcal{M}_{id}$, so does $\mathcal{G}_{exp}$ for expression mesh $\mathcal{M}_{exp}$. In order to control the distribution in latent space, we use variational strategy when training each branch. Let D_{id} and D_{exp} be the decoder for identity and expression extraction, and z_{id}, z_{exp} be the latent representation of each branch, the loss terms are defined as:

$$\begin{aligned}
L_{id} &= \|\mathcal{G}_{id} - D_{id}(z_{id})\|_1 \\
L_{id\_kld} &= KL(\mathcal{N}(0,1)\|Q(z_{id}|\mathcal{G}_{id})) \\
L_{exp} &= \|\mathcal{G}_{exp} - D_{exp}(z_{exp})\|_1 \\
L_{exp\_kld} &= KL(\mathcal{N}(0,1)\|Q(z_{exp}|\mathcal{G}_{exp})),
\end{aligned} \tag{3}$$

where L_{id} and L_{id_kld} are identity reconstruction loss and KullbackLeibler (KL) divergence loss, so do L_{exp} and L_{exp_kld} for expression attribute. The KL loss enforces a unit Gaussian prior $\mathcal{N}(0,1)$ with zero mean on the distribution of latent vectors $Q(z)$.

3.3. Fusion Network

As a representation, it is essential to rebuild the original input from the decomposed identity and expression attributes. Therefore, we naturally propose a fusion module to merge identity and expression meshes pair ($\mathcal{M}_{id}$, $\mathcal{M}_{exp}$) for reconstruction. And this module further guarantees that our decomposition is, in a sense, lossless. Since the mesh triplets are isomorphic, we can get a new graph by concatenating vertex features from identity and expression mesh. The new graph has the same edge set and vertex set with the original input, except for the concatenated $2d$-dimension feature on each vertex. The fusion module targets to convert this new graph with vertex feature in $\mathbb{R}^{2d}$ to an isomorphic graph with vertex feature in $\mathbb{R}^{d}$ (original input). We also apply spectral graph convolution with activation layers to achieve this target.

Now, let $\mathcal{G}_{cat} = [\hat{\mathcal{G}}_{id}, \hat{\mathcal{G}}_{exp}]$ be the concatenated new graph feature and $\mathcal{G}_{ori}$ be the feature of the original mesh $\mathcal{M}$. Here $\hat{\mathcal{G}}_{id}, \hat{\mathcal{G}}_{exp}$ are outputs of the identity/expression branch respectively. The loss function for the fusion module is:

$$L_{rec} = \|F(\mathcal{G}_{cat}) - \mathcal{G}_{ori}\|_1, \tag{4}$$

where F represents the fusion network.

3.4. Training Process

We first pretrain the decomposition network and fusion network sequentially. Then we train the entire network in an end-to-end strategy. During the end-to-end training step, we add disentangling loss in the following form:

$$L_{dis} = \|D_{exp}(E_{exp}(\hat{\mathcal{G}}_{id})) - \bar{\mathcal{G}}\|_1 + \|D_{id}(E_{id}(\hat{\mathcal{G}}_{exp})) - \bar{\mathcal{G}}\|_1, \tag{5}$$

where $\bar{\mathcal{G}}$ is the feature of mean neutral face, as shown in Fig. 2. The disentangling loss guarantees the identity part containing no expression information, and the expression part does not contain any identity information. In summary, the full loss function is defined as follow:

$$\begin{aligned} L_{total} = L_{rec} + L_{dis} + L_{id} + L_{exp} + \\ \alpha_{id\_kld} L_{id\_kld} + \alpha_{exp\_kld} L_{exp\_kld}. \end{aligned} \tag{6}$$

Data Augmentation We train our model with FaceWareHouse [14] dataset, which includes 150 identities and 47 expressions for each identity. In our experiment, as the quantity of identities is very small, there exists an over-fitting problem in the training process of identity decomposition branch. We develop a novel data augmentation method to overcome such over-fitting problem. Given m identity samples in the training set, we generate new 3D face meshes via interpolations among m samples. The deformation representation(DR) features of these identity samples are denoted as $(\mathbf{DR}_1, \mathbf{DR}_2, \ldots, \mathbf{DR}_m)$. We generate new DR features and reconstruct the 3D face meshes from these new DR features. We create an uniform distribution vector, $(r, \theta_1, \ldots, \theta_{m-1})$ in polar coordinates system, where r follows uniform distribution $\mathbf{U}(0.5, 1.2)$, and others follow uniform distribution $\mathbf{U}(0, \pi/2)$. We convert the above polar coordinates into Cartesian coordinates $(a_1, \ldots, a_m)$, and interpolate the sampled m DR features by $\sum_{i=1}^{m} a_i \mathbf{DR}_i$. These m features are a bootstrap sample from the training dataset. This data augmentation method can create various 3D faces with only several samples from the training set and can solve the over-fitting problem. In our experiment, we set $m = 5$ and generate 2000 new 3D face meshes (see supplementary for some examples) for training.

4. Experiment

In this section, we will first introduce our implementation[1] details in 4.1. Then we will introduce several metrics used for measuring reconstruction and decomposition accuracy in 4.2. Finally, we will show our experiments on two different datasets in Sec 4.3 and 4.4, including ablation study and comparison with baselines.

4.1. Implementation Details

At first, we introduce data preparation procedure of generate the ground-truth identity and expression mesh. Taking FaceWareHouse for example, the neutral expression of a subject represents his/her identity mesh. As for expression mesh, we compute the average shape of the same expression belonging to 140 subjects and define the output 47 expressions as the ground-truth meshes on *mean* face. These operations can also be applied to other 3D face shape data sets.

Our algorithm is implemented in Keras [16] with Tensorflow [1] backend. All the training and testing experiments were tested on a PC with NVIDIA TiTan XP and CUDA 8.0.

We train our networks for 50 epochs per step with a learning rate of 1e-4, and a learning rate decay of 0.6 every 10 epochs. The hyper-parameters $\alpha_{id\_kld}, \alpha_{exp\_kld}$ are set as 1e-5.

4.2. Evaluation Metric

The main target of our method is to decompose a given 3D face shape into identity and expression parts as accurate as possible and achieve high 3D shape reconstruction accuracy at the same time. Therefore, evaluation criteria are designed based on these two aspects.

4.2.1 Reconstruction Measurement

We adopt two kinds of metrics to evaluate the 3D shape reconstruction accuracy.

Average vertex distance The average vertex distance $\mathbf{E}_{avd}$ between reconstructed mesh $\mathcal{M}'$ and original mesh $\mathcal{M}$ is defined as:

$$\mathbf{E}_{avd}(\mathcal{M}, \mathcal{M}') = \frac{1}{|\mathcal{V}|} \sum_{i=1}^{|\mathcal{V}|} \|v_i - v_i'\|_2. \tag{7}$$

Perceptual Error As E_{avd} can not reflect perceptual distance [17, 39]. In [39], spatial-temporal edge difference was proposed to measure perceptual distance by the local error of dynamic mesh independent of its absolute position. In this work, we adopt the spatial edge difference error $\mathbf{E}_{sed}$ to measure the perceptual error. Let e_{ij} be the edge connects v_i and v_j of original mesh $\mathcal{M}$, and edge e_{ij}' is the corresponding edge in reconstructed mesh $\mathcal{M}'$, the relative edge difference is defined as:

$$ed(e_{ij}, e_{ij}') = \left|\frac{\|e_{ij}\| - \|e_{ij}'\|}{\|e_{ij}\|}\right| \tag{8}$$

The weighted average of relative edge difference around a vertex v_i is computed as:

$$\bar{ed}(v_i) = \frac{\sum_{j \in \mathcal{N}_i} l_{ij} ed(e_{ij}, e_{ij}')}{\sum_{j \in \mathcal{N}_i} l_{ij}},$$

where l_{ij} is the edge length of edge e_{ij}. Therefore the local deviation around a vertex v_i can be expressed by

$$\sigma(v_i) = \sqrt{\frac{\sum_{j \in \mathcal{N}_i} l_{ij} (ed(e_{ij}, e_{ij}') - \bar{ed}(v_i))^2}{\sum_{j \in \mathcal{N}_i} l_{ij}}}. \tag{9}$$

[1] Avalible at https://github.com/zihangJiang/DR-Learning-for-3D-Face

Method	E_{avd}		E_{sed}		E_{id}		E_{exp}	
	Mean Error	Median	Mean Error	Median	Mean Error	Median	Mean Error	Median
Bilinear [14]	0.993	0.998	0.0243	0.0183	0.477	0.472	0.527	0.484
FLAME [26]	0.882	0.905	0.0144	0.0074	0.329	0.328	0.711	0.630
MeshAE [3]	0.825	0.811	0.0151	0.0777	-	-	-	-
Ours w/o DR & Fusion	0.981	1.292	0.177	0.0938	0.395	0.380	0.170	0.160
Ours w/o DR	0.939	0.836	0.447	0.388	0.446	0.463	0.0992	0.0750
Ours w/o Fusion	0.661	0.579	**0.00283**	**0.0000**	0.183	0.178	0.0582	0.0494
Ours	**0.472**	**0.381**	0.00333	**0.0000**	**0.121**	**0.121**	**0.0388**	**0.0267**

Table 1. Quantitative results on Facewarehouse. All number were in millimeters. DR: deformation representation; Fusion: fusion module.

We compute the average local deviation over all the vertices and get the spatial edge difference error:

$$\mathbf{E}_{sed} = \frac{1}{|\mathcal{V}|}\sum_{i=1}^{|\mathcal{V}|}\sigma(v_i). \tag{10}$$

And smaller value of $\mathbf{E}_{sed}$ means better perceptual result.

4.2.2 Decomposition Measurement

To measure the disentangled representation for 3D face shape, we propose a metric for reconstructed *identity mesh* from the models with the same identity and different expressions, and *expression mesh* from the models with different identities and the same expression.

Taking identity part for example, we denote $\{\mathcal{M}^i\}$ as the test set containing a series of expressions of an identical person. A good decomposition method is supposed to decompose $\{\mathcal{M}^i\}$ into several similar identity features and various expression features. Moreover, the meshes reconstructed from those identity features are supposed to be similar with each other, hence the standard deviation of reconstructed identity meshes $\{\mathcal{M}^i_{id}\}$ is suitable to be used to evaluate the decomposed ability of the disentangled representation. And it is the same to other test set $\{\mathcal{N}^j\}$ consisted of identical expressions and different identities. So the decomposition metric is defined as follow:

$$\begin{aligned}\mathbf{E}_{id} &= \sigma(\{\mathcal{M}^i_{id}\}) \\ \mathbf{E}_{exp} &= \sigma(\{\mathcal{N}^j_{exp}\}),\end{aligned} \tag{11}$$

where $\{\mathcal{M}^i_{id}\}$ and $\{\mathcal{N}^j_{exp}\}$ are reconstructed identity and expression meshes of test sets $\{\mathcal{M}^i\}$ and $\{\mathcal{N}^j\}$, while σ is the standard deviation operator. This metric adopts vertex distance.

4.3. Experiments on FaceWareHouse [14]

FaceWareHouse is a widely used 3D face shape dataset developed by Cao *et al.*, which includes 47 expressions along 150 different identities. It is easy to obtain the training triplets from Facewarehouse dataset. We conduct ablation study of our framework and compare our method with the bilinear model which is widely referred with this dataset. In all the experiments of this part, we choose the first 140 identities with their expression face shapes for training, and the left 10 identities for testing.

4.3.1 Baseline Comparison

Bilinear model Cao *et al.* [14] proposed 2-mode tensor product formulation for 3D face shape representations as:

$$\mathcal{M} = C_r \times_2 \boldsymbol{\alpha}_{id} \times_3 \boldsymbol{\alpha}_{exp} \tag{12}$$

where C_r is the reduced core tensor containing the top-left corner of the original tensor produced by HO-SVD decomposition, $\boldsymbol{\alpha}_{id}$ and $\boldsymbol{\alpha}_{exp}$ are the row vectors of identity and expression weights. And 50 and 25 are recommended as the proper reduced dimensions of identity and expression subspaces [14].

For a given 3D face shape, $\boldsymbol{\alpha}_{id}$ and $\boldsymbol{\alpha}_{exp}$ can be optimized by applying *Alternating Least Squares* (ALS) method to the tensor contraction. We denote $\{\mathcal{M}^i\}$ like we used in 4.2.2 and optimize $(\boldsymbol{\alpha}^i_{id}, \boldsymbol{\alpha}^i_{exp})$ for each $\mathcal{M}^i$. The *identity mesh* is reconstructed with identity parameters $\boldsymbol{\alpha}^i_{id}$ and neutral expression parameters, and the *expression mesh* is reconstructed with mean face identity and expression parameter $\boldsymbol{\alpha}^i_{exp}$.

FLAME Li *et al.* [26] propose FLAME model by representing 3D face shape including identity, expression, head rotation, and yaw motion with linear blendskinning and achieve state of the art result. For comparison, we train FLAME with identity model and expression model.

MeshAE Anurag [3] proposed a spectral graph convolutional mesh autoencoders (MeshAE) structure for 3D face shape embedding. We also evaluate the model's reconstruction ability on FaceWareHouse dataset as it encode whole shape 3D face without disentangling identity and expression.

For a fair comparison, the dimensions of our latent spaces (identity z_{id} and expression z_{exp}) are separately set as 50 and 25, the same with the bilinear model and FLAME. And the size of latent space for Mesh AutoEncoder (MeshAE) is set as 75. Quantitative results are given in Tab

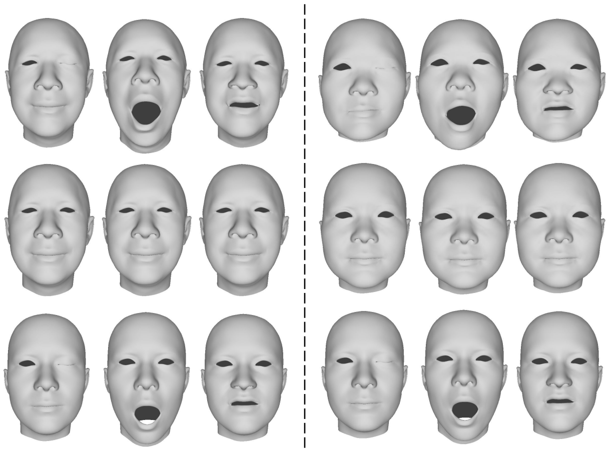


Figure 3. Results of identity and expression decomposition. The original and extracted identity and expression components are given from top to bottom. We show samples from two subjects.

1. Our framework gets much better result in each evaluation. We also show qualitative visual result of our results on identity and expression decomposition in Fig 3. The visual result and numerical result demonstrate that our disentangled learning not only achieves better reconstruction accuracy but also neatly decouples expression and identity attributes.

4.3.2 Ablation Study

In our framework, we have two novel designs including 3D face shape representation and fusion network, which greatly improve the representation ability of our method. To investigate the effectiveness of these two designs, Tab. 1 presents the variants of our learning method, where w/o is the abbreviation of without. In the following, we compare our well-designed framework with other implementation strategies.

We adopt a novel vertex based deformation representation [21] for 3D face shape. Another straightforward way is to directly use the Euclidean coordinates as the method in [3]. The results of without using DR is reported in Tab. 1.

Another novel design in our pipeline is the fusion network. A natural replacement for fusion module is to represent 3D face as a composite model like 3DMM [5, 27]:

$$\mathcal{G} = \bar{\mathcal{G}} + D_{id}(z_{id}) + D_{exp}(z_{exp}). \tag{13}$$

where $\bar{\mathcal{G}}$ is the feature of mean face. The result that without using fusion is shown in Tab. 1. We also report errors without using both designs. It can be observed from the ablation study, both DR and fusion network greatly improve the performance. DR significantly improved our model's performance in the average vertex distance error evaluation. And the fusion module helps to disentangle the expression more naturally *i.e*. achieves smaller error in E_{exp}. Our proposed framework get a slightly higher error in E_{sed} when adding the fusion module. While considering for all evaluation metrics, our method still achieves more satisfying result than other comparative tests.

Average error	Mean Error	Median Error
FLAME [26]	2.001	1.615
Ours	**1.643**	**1.536**

Table 2. Extrapolation results on COMA dataset. All results are in millimeters.

4.4. Experiment on COMA Dataset [3]

Very recently, Anurag *et al*. released the COMA dataset which includes 20,466 3D face models. This dataset is captured at 60fps with a multi-camera active stereo system, which contains 12 identities performing 12 different expressions. COMA dataset was used to build a nonlinear 3D face representation [3], while it encodes and decodes the whole 3D face shape into one vector in the latent space without considering identity and expression attribute. We evaluate the ability of extrapolation over expression by training our model with COMA dataset. However, different from FaceWareHouse dataset, the shape models in COMA dataset are not specified with expression labels. We manually select 12 models with representative expressions for all the 12 identities. For each shape model in the remaining, the residual DR feature between the original model and its identity model is used for supervision during the training process.

To measure the generalization of our model, we perform 12 cross validation for one expression. For our method, we set our latent vector size as 8, with 4 for identity and 4 for expression. And we compare our method with FLAME, which is the state-of-the-art 3D face model representation with decomposed attributes. For comparison, FLAME is trained for expression model and obtained with 8 components for identity and expression respectively.

We compare our method with FLAME on expression extrapolation experiment, and report the average vertex distance as defined in Eq. (7) on all the 12 cross validation experiments in Tab. 2. It can be observed that our method gets better generalization result compared with the state-of-the-art FLAME method on extrapolation experiment. All the 12 expressions extrapolation cross validation experiments are given in supplementary.

4.5. Discussion on Larger Dataset

There is a long-standing problem in conducting learning method in 3D vision topic, which is lack of 3D data. Recently, more and more methods proposed solution to tackle this problem, *e.g*. combine multiple dataset by non-rigid registration. In our framework, we adopt a novel data

Dataset	E_{avd}	E_{sed}	E_{id}	E_{exp}
Original FWH	18.3/18.0	0.05/0.03	1.4/1.4	0.5/0.3
Combination	16.9/16.6	0.06/0.03	1.6/1.6	0.5/0.4
DR-augmented	4.7/3.8	0.03/0.00	1.2/1.2	0.4/0.3

Table 3. More quantitative results. Table gives our results on different datasets: original FWH, combination of Bosphorus and FWH (Combination) and our DR-augmented FWH. All number in 0.1 millimeters.

augmentation strategy by interpolation/extrapolation of DR feature. We also design an experiment on a large-scale dataset. We create a larger dataset by convert Bosphorus [32] to mesh by nonrigid registration and combine with FaceWareHouse. We evaluate our method on three different training datasets: original FaceWareHouse, combination of FaceWareHouse and Bosphorus, and DR-augmented FWH. Tab. 3 shows the comparison results. Our augmentation strategy leads to the best scores on all aspects, which demonstrates that it greatly improves the model's stability and robustness. We hope our data augmentation strategy can benefit 3D vision community.

5. Application

Based on our proposed disentangled representation for 3D face shape, we can apply our model in many applications like expression transfer and face recognition. In the following part, we first show that our method can achieve better performances than traditional method on expression transfer, and then we show the shape exploration results in the trained identity and expression latent space of our model.

5.1. Expression Transfer

A standard solution for expression transfer [40, 12, 36] is to transfer the expression weights from source to target face. We randomly select two identities from the test data set of FaceWareHouse to compare the expression transfer results of the bilinear model and our method. For the bilinear model, we first solve the identity and expression parameters for the reference model and then transfer the expression parameter from the source to the target face. In our method, we directly apply the latent expression code of source face to the target face. Some results are shown in Fig. 5. The corresponding expressions on the target object in FaceWareHouse dataset are treated as the ground truth. It can be easily observed that our method can achieve more natural and accurate performances, and our results are closer to the ground truth in quantitative error evaluations.

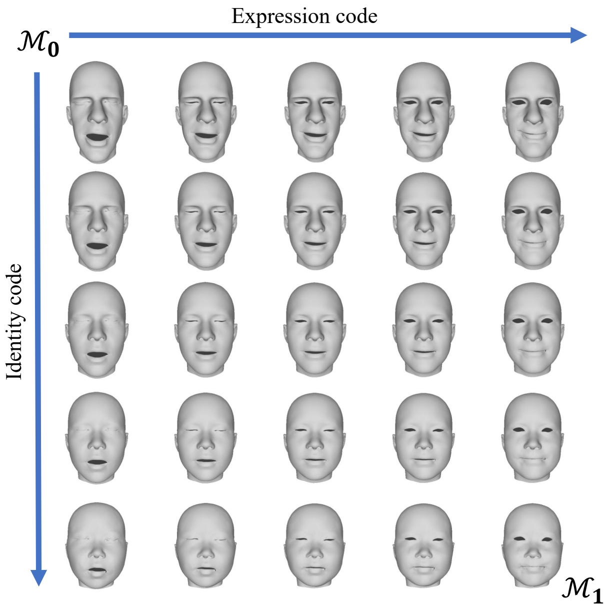


Figure 4. Exploring interpolation results on latent space. Based on our method, we can obtain identity and expression code for two 3D face model $\mathcal{M}_0$ and $\mathcal{M}_1$, and we interpolate latent identity and expression vectors individually, in stride of 0.25.

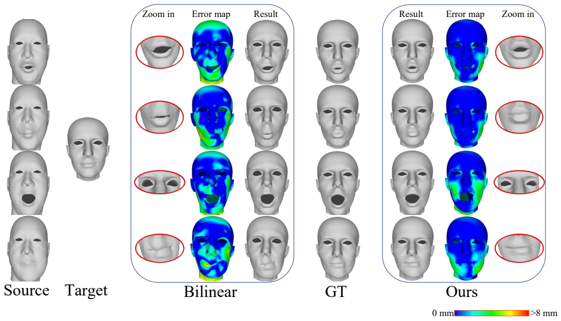


Figure 5. Expression transfer application. Comparing to the bilinear model, our method achieves more natural and stable visual results.

5.2. Latent space interpolation

Our disentangled representation includes two latent codes for identity and expression. With the learned latent spaces, we can interpolate models by gradually changing identities and expressions. The interpolating operation is applied on the latent code, and the models are recovered from the generated code with the trained decoder. In this experiment, We interpolate latent code by step of 0.25 in identity and expression separately, and thus we can observe that the interpolation results are meaningful and reasonable as shown in Fig. 4,

6. Conclusion

We have proposed a disentangled representation learning method for 3D face shape. A given 3D face shape can be accurately decomposed into identity part and expression part. To effectively solve this problem, a well-designed framework is proposed to train decomposition networks and fusion network. To better represent the non-rigid deformation space, the input face shape is represented as vertex based deformation representation rather than Euclidean coordinates. We have demonstrated the effectiveness of the proposed method via ablation study and extensive quantitative and qualitative experiments. Applications like expression transfer based on our disentangled representation have shown more natural and accurate results compared with traditional method.

References

[1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. Tensorflow: a system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016. 5

[2] B. Amberg, R. Knothe, and T. Vetter. Expression invariant 3d face recognition with a morphable model. In *Automatic Face & Gesture Recognition, 2008. FG'08. 8th IEEE International Conference on*, pages 1–6. IEEE, 2008. 1

[3] S. S. Anurag Ranjan, Timo Bolkart and M. J. Black. Generating 3D faces using convolutional mesh autoencoders. In *European Conference on Computer Vision (ECCV)*, 2018. 1, 2, 3, 6, 7, 11

[4] T. Bagautdinov, C. Wu, J. Saragih, P. Fua, and Y. Sheikh. Modeling facial geometry using compositional vaes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2

[5] 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 1999, Los Angeles, CA, USA, August 8-13, 1999*, pages 187–194, 1999. 1, 2, 3, 7

[6] J. Booth, E. Antonakos, S. Ploumpis, G. Trigeorgis, Y. Panagakis, S. Zafeiriou, et al. 3d face morphable models in-the-wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2

[7] J. Booth, A. Roussos, A. Ponniah, D. Dunaway, and S. Zafeiriou. Large scale 3d morphable models. *International Journal of Computer Vision*, 126(2-4):233–254, 2018. 1

[8] J. Booth, A. Roussos, S. Zafeiriou, A. Ponniah, and D. Dunaway. A 3d morphable model learnt from 10,000 faces. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5543–5552, 2016. 1

[9] M. Botsch and O. Sorkine. On linear variational surface deformation methods. *IEEE Transactions on Visualization and Computer Graphics*, 14(1):213–230, 2008. 3

[10] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017. 2

[11] J. Bruna, W. Zaremba, A. Szlam, and Y. Lecun. Spectral networks and locally connected networks on graphs. In *International Conference on Learning Representations*, 2014. 2, 3

[12] C. Cao, Q. Hou, and K. Zhou. Displaced dynamic expression regression for real-time facial tracking and animation. *ACM Transactions on graphics (TOG)*, 33(4):43, 2014. 1, 8

[13] C. Cao, Y. Weng, S. Lin, and K. Zhou. 3d shape regression for real-time facial animation. *ACM Transactions on Graphics (TOG)*, 32(4):41, 2013. 1

[14] 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*, 20(3):413–425, 2014. 1, 2, 3, 5, 6

[15] Y. Chang, C. Hu, R. Feris, and M. Turk. Manifold based analysis of facial expression. *Image and Vision Computing*, 24(6):605–614, 2006. 2

[16] F. Chollet et al. Keras. https://github.com/fchollet/keras, 2015. 5

[17] M. Corsini, M.-C. Larabi, G. Lavoué, O. Petřík, L. Váša, and K. Wang. Perceptual metrics for static and dynamic triangle meshes. In *Computer Graphics Forum*, volume 32, pages 101–125. Wiley Online Library, 2013. 5

[18] M. Defferrard, X. Bresson, and P. Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems*, pages 3844–3852, 2016. 2, 3

[19] P. Ekman. Facial action coding system (facs). *A human face*, 2002. 2, 3

[20] K. Fisher, J. R. Towler, and M. Eimer. Facial identity and facial expression are initially integrated at visual perceptual stages of face processing. *Neuropsychologia*, 80:115–125, 2016. 1

[21] L. Gao, Y.-K. Lai, J. Yang, L.-X. Zhang, L. Kobbelt, and S. Xia. Sparse data driven mesh deformation. *arXiv preprint arXiv:1709.01250*, 2017. 2, 3, 7

[22] A. E. Ichim, S. Bouaziz, and M. Pauly. Dynamic 3d avatar creation from hand-held video input. *ACM Transactions on Graphics (TOG)*, 34(4):45, 2015. 1

[23] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017. 3

[24] P. Koppen, Z.-H. Feng, J. Kittler, M. Awais, W. Christmas, X.-J. Wu, and H.-F. Yin. Gaussian mixture 3d morphable face model. *Pattern Recognition*, 74:617–628, 2018. 1

[25] Q. Li, Z. Han, and X.-M. Wu. Deeper insights into graph convolutional networks for semi-supervised learning. *arXiv preprint arXiv:1801.07606*, 2018. 3

[26] T. Li, T. Bolkart, M. J. Black, H. Li, and J. Romero. Learning a model of facial shape and expression from 4D scans. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6), 2017. 1, 2, 3, 6, 7, 11, 15

[27] F. Liu, R. Zhu, D. Zeng, Q. Zhao, and X. Liu. Disentangling features in 3d face shapes for joint face reconstruction and recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2, 7

[28] M. Lüthi, T. Gerig, C. Jud, and T. Vetter. Gaussian process morphable models. *IEEE transactions on pattern analysis and machine intelligence*, 40(8):1860–1873, 2018. 1

[29] J. Masci, D. Boscaini, M. Bronstein, and P. Vandergheynst. Geodesic convolutional neural networks on riemannian manifolds. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 37–45, 2015. 2

[30] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[31] T. Papatheodorou and D. Rueckert. 3d face recognition. In *Face Recognition*. InTech, 2007. 1

[32] A. Savran, N. Alyüz, H. Dibeklioğlu, O. Çeliktutan, B. Gökberk, B. Sankur, and L. Akarun. Bosphorus database for 3d face analysis. In *European Workshop on Biometrics and Identity Management*, pages 47–56. Springer, 2008. 8

[33] A. Sinha, J. Bai, and K. Ramani. Deep learning 3d shape surfaces using geometry images. In *European Conference on Computer Vision*, pages 223–240. Springer, 2016. 2

[34] Q. Tan, L. Gao, Y. Lai, J. Yang, and S. Xia. Mesh-based autoencoders for localized deformation component analysis. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 2452–2459, 2018. 2

[35] J. Thies, M. Zollhöfer, M. Nießner, L. Valgaerts, M. Stamminger, and C. Theobalt. Real-time expression transfer for facial reenactment. *ACM Trans. Graph.*, 34(6):183:1–183:14, 2015. 1

[36] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2387–2395, 2016. 1, 8

[37] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner. Facevr: Real-time gaze-aware facial reenactment in virtual reality. *ACM Trans. Graph.*, 37(2):25:1–25:15, 2018. 1

[38] L. Tran and X. Liu. Nonlinear 3d face morphable model. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2

[39] L. Vása and V. Skala. A perception correlated comparison method for dynamic meshes. *IEEE Transactions on Visualization and Computer Graphics*, 17:220–230, 2011. 5

[40] D. Vlasic, M. Brand, H. Pfister, and J. Popović. Face transfer with multilinear models. *ACM transactions on graphics (TOG)*, 24(3):426–433, 2005. 2, 8

[41] Q. Wu, J. Zhang, Y.-K. Lai, J. Zheng, and J. Cai. Alive caricature from 2d to 3d. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 3

[42] F. Yang, J. Wang, E. Shechtman, L. Bourdev, and D. Metaxas. Expression flow for 3d-aware face component transfer. *ACM Transactions on Graphics (TOG)*, 30(4):60, 2011. 1

[43] X. Zhu, Z. Lei, J. Yan, D. Yi, and S. Z. Li. High-fidelity pose and expression normalization for face recognition in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 787–796, 2015. 1

A. Network Structure

Our network structure is shown in Fig. 6, and we choose Chebyshev polynomials of order 2 as hyper-parameter of our convolution layers. During training process, we duplicate identity and expression branch, respectively, to get the disentangling loss L_{dis} as given in Sec.3.4.

B. Latent space dimension exploration

In our paper, we compare our model ability with other baseline models on FaceWareHouse with latent space size is 25 for expression and 50 for identity. We also conduct experiment about our method with different size of latent space. The result shown in Tab. 4.

method	E_{avd}	E_{sed}	E_{id}	E_{exp}
Ours (25/10)	6.7/5.9	0.06/0.02	1.3/1.3	0.4/0.3
Ours (75/50)	3.7/2.8	0.02/0.00	0.9/0.9	0.3/0.2
Ours (50/25)	4.7/3.8	0.03/0.00	1.2/1.2	0.4/0.3

Table 4. More quantitative results. Ours$(25/10)$ represents that identity latent dim is set to 25 and expression latent dim is set to 10. So do Ours(75/50) and original result, Ours(50/25). All number in 0.1 millimeters.

C. Deformation Representation Reconstruction Accuracy

We use deformation representation in our framework, and the conversion from deformation representation to 3D mesh is solved by a least-square problem. We compute the geometric distance between original point clouds and DR-reconstructed ones over FaceWarehouse, and the average error is 31 micrometers. It means that the conversation process has very little influence on reconstruction accuracy.

D. Data Augmentation Samples

As descripted in Sec.3.4, we augment 2000 meshes with neutral expression from the FaceWareHouse dataset for identity decomposition branch training, and Fig. 7 shows some examples from the augmented models.

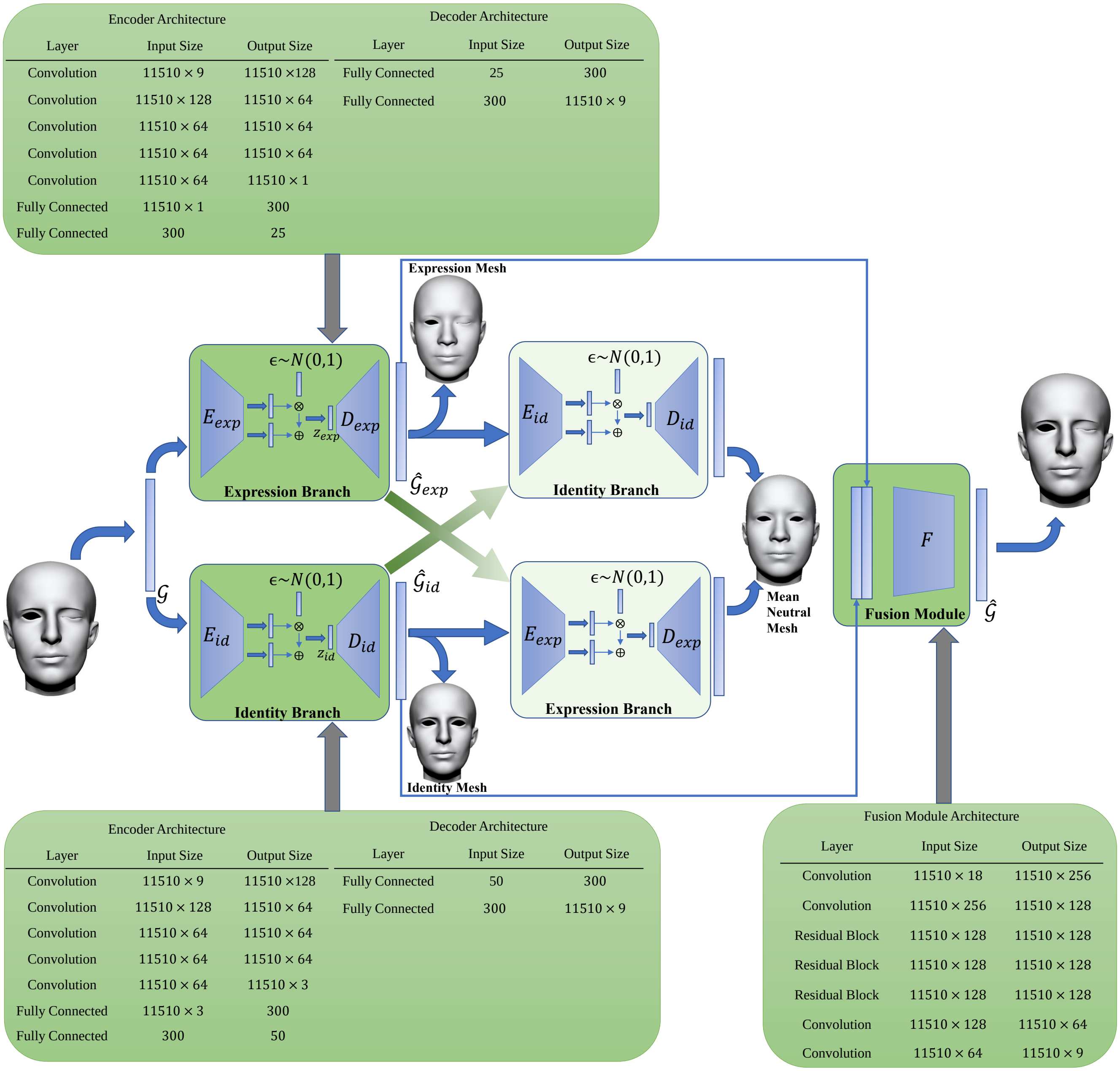
E. COMA Dataset [3]

E.1. Selected Expressions from COMA Dataset

In Fig 8, we show our selected 144 expressions from COMA dataset [3] for our decomposition and fusion networks pretraining in Sec.4.4. Each column is of the same identity with 12 various expressions.

E.2. 12 Cross Validation Experiment Result

We show the numerical result of 12 cross validation experiments compared with FLAME [26] in Tab 5. Our method gets lower error in most cases. For some case like bareteeth, our method gets higher median error than FLAME. Most error of our method is caused by the bias resulting from manual selection on expressions.



Encoder Architecture

Layer	Input Size	Output Size
Convolution	11510 × 9	11510 ×128
Convolution	11510 × 128	11510 × 64
Convolution	11510 × 64	11510 × 64
Convolution	11510 × 64	11510 × 64
Convolution	11510 × 64	11510 × 1
Fully Connected	11510 × 1	300
Fully Connected	300	25

Decoder Architecture

Layer	Input Size	Output Size
Fully Connected	25	300
Fully Connected	300	11510 × 9

Encoder Architecture

Layer	Input Size	Output Size
Convolution	11510 × 9	11510 ×128
Convolution	11510 × 128	11510 × 64
Convolution	11510 × 64	11510 × 64
Convolution	11510 × 64	11510 × 64
Convolution	11510 × 64	11510 × 3
Fully Connected	11510 × 3	300
Fully Connected	300	50

Decoder Architecture

Layer	Input Size	Output Size
Fully Connected	50	300
Fully Connected	300	11510 × 9

Fusion Module Architecture

Layer	Input Size	Output Size
Convolution	11510 × 18	11510 × 256
Convolution	11510 × 256	11510 × 128
Residual Block	11510 × 128	11510 × 128
Residual Block	11510 × 128	11510 × 128
Residual Block	11510 × 128	11510 × 128
Convolution	11510 × 128	11510 × 64
Convolution	11510 × 64	11510 × 9

Figure 6. Our Network Structure.

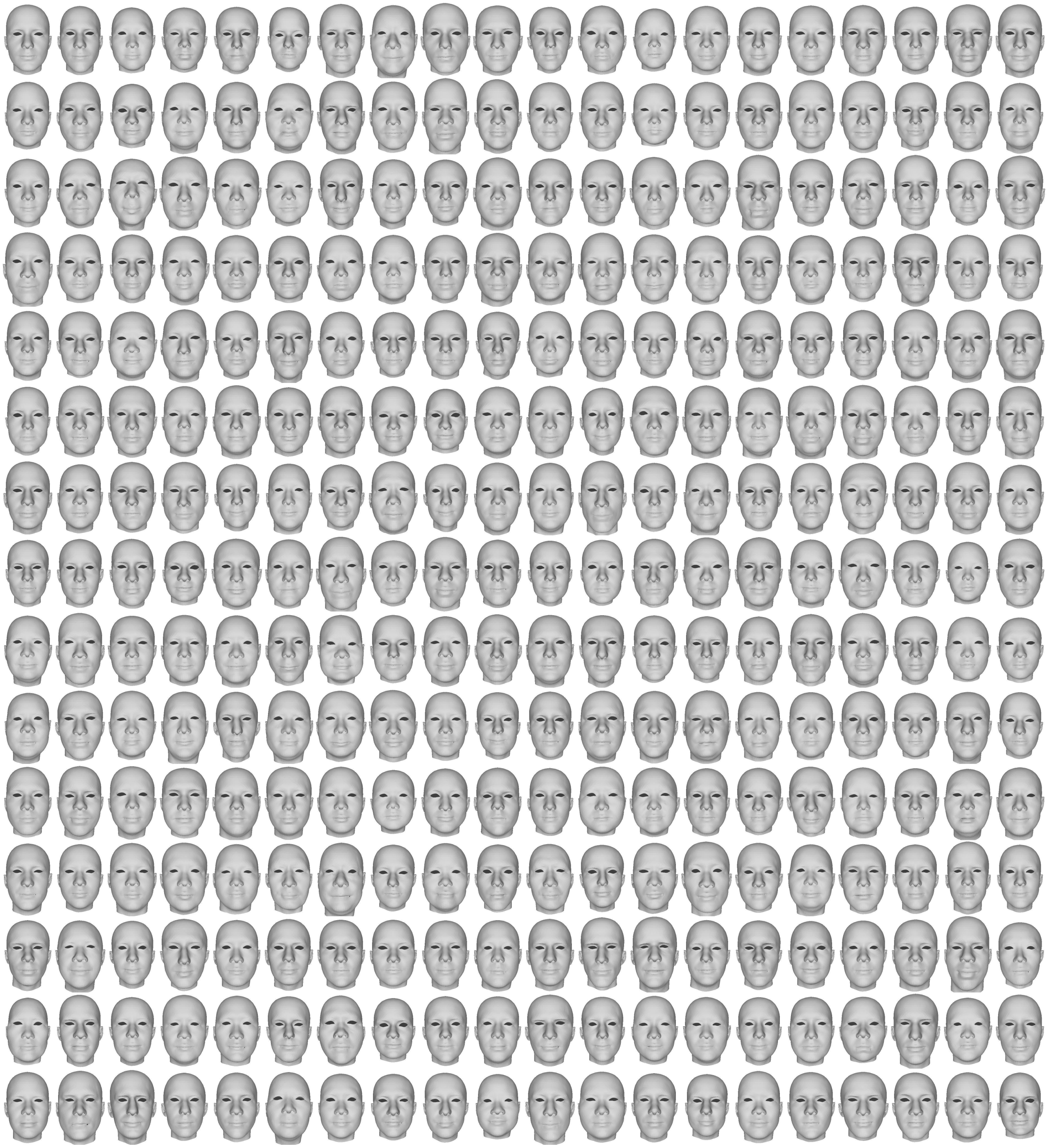


Figure 7. Data augmentation samples.

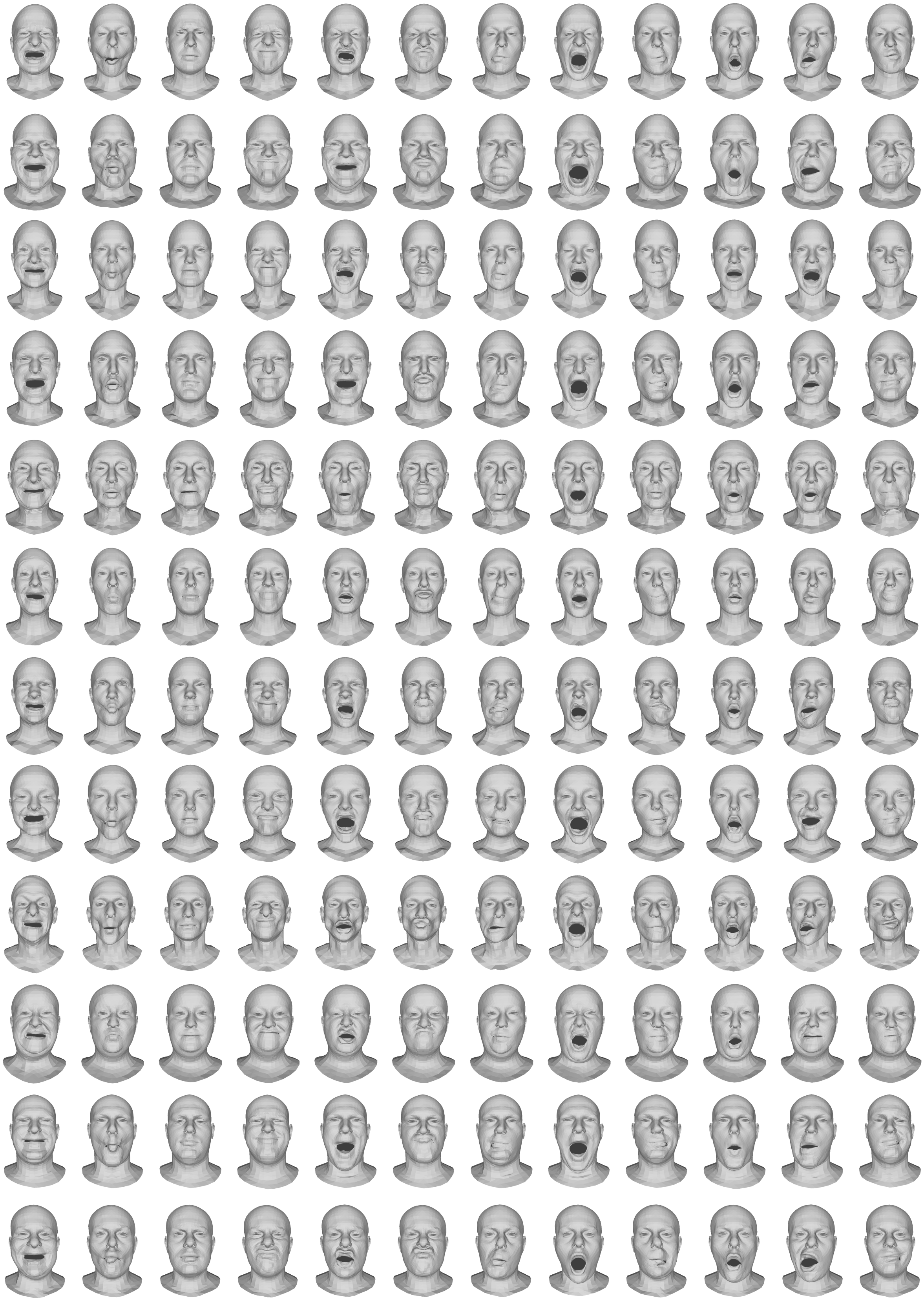


Figure 8. Selected 144 expressions from COMA dataset.

	Ours		FLAME [26]	
	Mean Error	Median	Mean Error	Median
bareteeth	**1.695**	1.673	2.002	**1.606**
cheeks in	**1.706**	**1.605**	2.011	1.609
eyebrow	**1.475**	**1.357**	1.862	1.516
high smile	**1.714**	1.641	1.960	**1.625**
lips back	**1.752**	**1.457**	2.047	1.639
lips up	**1.747**	**1.515**	1.983	1.616
mouth down	**1.655**	**1.587**	2.029	1.651
mouth extreme	**1.551**	**1.429**	2.028	1.613
mouth middle	**1.757**	1.691	2.043	**1.620**
mouth open	**1.393**	**1.371**	1.894	1.544
mouth side	**1.748**	**1.610**	2.090	1.659
mouth up	**1.528**	**1.499**	2.067	1.680
Average	**1.643**	**1.536**	2.001	1.615

Table 5. Comparison between our method and FLAME [26] on expression extrapolation experiment by testing on COMA dataset. Errors are in millimeters.