3D FACE REPRENTATION AND RECONSTRUCTION WITH MULTI-SCALE GRAPH CONVOLUTIONAL AUTOENCODERS

Cunkuan Yuan [1], *Kun Li* [1*], *Yu-Kun Lai* [2], *Yebin Liu* [3], *Jingyu Yang* [1]

[1] Tianjin University, Tianjin, China
[2] Cardiff University, Wales, UK
[3] Tsinghua University, Beijing, China

ABSTRACT

Effective representation and reconstruction for human faces are very important in many applications. Existing linear representation methods cannot reconstruct high quality 3D faces with details, while the newest non-linear representation method is less suitable for real shapes since spectral decompositions are unstable across different graphs. To address these problems, we propose a multi-scale graph convolutional autoencoder for face representation and reconstruction. Our autoencoder uses graph convolution, which is easily trained for the data with graph structures and can be used for other deformable models. Our model can also be used for variational training to generate high quality face shapes. Experimental results demonstrate that our model can generate more plausible, complex, and stable 3D shapes, and achieves higher quality face reconstruction compared with state-of-the-art methods.

***Index Terms*—** Face representation, face reconstruction, autoencoder, mesh, variational training

1. INTRODUCTION

Human faces play a key role in identity recognition, message transmission, and emotional expression. Effective representation and reconstruction of a specific face are very important for creating personalized avatars, 3D printing, and face animation, which have a wide range of applications in movie production, computer games, augmented reality (AR) and virtual reality (VR). However, human faces are highly variable as they are affected by many factors such as age, sex, ethnicity, *etc*., and deform significantly with expressions. Therefore, it is difficult to effectively represent and reconstruct such non-linear deformations.

Traditional methods use a laser scanner or a depth camera to reconstruct a 3D face using fusion-based methods [1], but they cannot achieve animation, editing and generation. To address this problem, parametric face models [2, 3, 4] and blendshapes [5] are proposed to represent facial shapes and expressions, and several methods successfully reconstruct the face shape from the scanned depth mesh using these models [6, 7]. However, the reconstructed shapes using the linear representation are generally smooth without rich details.

Deep learning has great success in many application areas, especially convolutional neural networks (CNN). Volumetric representations [8] and point clouds [9] are used to reconstruct 3D models using CNN, but these methods require a lot of memory and cannot effectively represent face deformations. The surface mesh, as a common representation for face shapes, has a graph structure with irregular connectivity that is difficult to directly use CNN. To better represent faces, Ranjan *et al*. [10] introduce a model that learns a non-linear representation of a face using spectral convolutions on their dataset containing $20,466$ high resolution face meshes with extreme facial expressions. However, this method uses spectral filtering to generalize convolutional network to irregular graph structured data, which is less suitable for real shapes since global decompositions are unstable across different graphs.

To address these problems, we propose a new face representation and reconstruction method with hierarchical graph convolutional mesh autoencoders (MAE), which can achieve higher quality reconstruction. Our MAE model is primarily composed of dynamic filtering convolutional layers [11], which dynamically compute the correspondences between filter weights and graph neighborhoods with arbitrary connectivity from features learned by the network. We also perform multi-scale sampling on the mesh to obtain a hierarchical face representation. Our MAE model can be used for variational training to generate high quality face shapes. Experimental results show that our method has higher accuracy of face reconstruction than state-of-the-art method [10], and can generate plausible high-quality facial meshes by variational training.

The main contributions of this work are summarized as:

- We propose a new graph convolutional autoencoder with a hierarchical multi-scale representation for face surface meshes. Our model relies on the vertex con-

*Corresponding author: lik@tju.edu.cn. This work was supported in part by the National Natural Science Foundation of China (Grant 61571322 and 61771339), and Tianjin Research Program of Application Foundation and Advanced Technology under Grant 18JCYBJC19200.

nection relationship of the mesh for convolution, and can also generate a hierarchical mesh representation by effectively sampling the vertices of the mesh.

- Our autoencoder uses mesh raw data without complex embedding process and is easily trained.
- We show that our method has higher reconstruction accuracy than state-of-the-art method and can generate more plausible, complex and stable 3D shapes.

2. RELATED WORK

2.1. Face Representation

Most methods use statistical priors to model the structure and expression of faces [6, 7]. 3D Morphable Model (3DMM) proposed by Blanz and Vetter [2] is the first and the most popular face representation prior based on principal component analysis (PCA). Basel Face Model (BFM) [12] is the publicly available version of this model and has been widely used in many applications. Booth *et al*. [3] proposed another linear face model learned from nearly 10,000 facial scans of more diverse subjects in a neutral expression. However, the above linear representation methods cannot capture extreme deformations and non-linear expressions. Ranjan *et al*. [10] propose a non-linear representation model learned by spectral convolutions on their dataset containing 20,466 high resolution meshes with extreme facial expressions. But this method uses spectral filtering to generalize convolutional network to irregular graph structured data, which is less suitable for real shapes since global decompositions are unstable across different graphs. In this paper, we propose a new model using dynamic filtering convolutional layers with a multi-scale representation to be more suitable for meshes.

2.2. Generative Modeling

Traditional methods [13, 14] use probabilistic inference for 3D model generation (synthesis), but they are only suitable for specific 3D shapes. Recent work considers using deep learning for 3D model generation. Wu *et al*. [8] propose a generative model using a convolutional deep belief network on a 3D voxel grid, but volumetric operations require a lot of memory and can only synthesize coarse 3D shapes without details. Tan *et al*. [15] use RIMD (Rotation Invariant Mesh Difference) representation and a variational autoencoder (VAE) on meshes to generate new shapes not existing in the original dataset. Gao *et al*. [16] propose a new variational autoencoder to encode shape deformations and a cycle-consistent generative adversarial network (GAN) for reliable mapping between latent spaces. However, these methods require dedicated deformation representations and may not represent subtle details well. In this paper, we use multi-scale mesh sampling operations combined with graph convolutions to better model high resolution details.

3. METHOD

In order to simultaneously achieve face latent representation and reconstruction, we propose a multi-scale mesh AE model. Note that our method can be extended to other deformable objects.

3.1. Multi-scale MAE Model

Define a 3D facial mesh (or a general deformable mesh) as a set of vertices and edges, $\mathcal{F} = (\mathbf{V}, \mathcal{N})$, with $|\mathbf{V}| = n$ vertices that lie in 3D Euclidean space, $\mathbf{V} \in \mathbb{R}^{n\times 3}$. The adjacency matrix $\mathcal{N}$ is a collection of edge sets that represents the neighborhood for each vertex. Our multi-scale mesh AE consists of two parts: an encoder and a decoder. The encoder encodes the 3D mesh $\mathcal{F}$ into a latent vector $\mathbf{z} = E(\mathcal{F})$, and the decoder decodes the latent vector into a 3D mesh $\mathcal{F} = D(\mathbf{z})$. Traditional CNNs cannot deal with such irregular data graphs, and thus we use dynamic filtering convolutional layers [11] to process the mesh data. It can learn the mapping from the neighborhood patch to filter weights, which considers the intrinsic characteristic of the mesh. Specifically, the input to a layer is a feature vector $\mathbf{x}_i$ associated with a vertex $i \in \{1, ..., n\}$, and the output is also a vector $\mathbf{y}_i$:

$$\mathbf{y}_i = \mathbf{b} + \sum_{m=1}^{M} \frac{1}{|\mathcal{N}_i|} \sum_{j\in\mathcal{N}_i} e_m(\mathbf{x}_i, \mathbf{x}_j)\mathbf{W}_m \mathbf{x}_j, \tag{1}$$

where $\mathcal{N}_i$ is the set of neighbors of vertex i, and $\{\mathbf{W}_m \in \mathbb{R}^{N_x\times N_y}\}$ is a set of M weight matrices for the filters. N_x and N_y are the dimensions of $\mathbf{x}_i$ and $\mathbf{y}_i$ respectively. $e_m(\mathbf{x}_i, \mathbf{x}_j) \propto \exp(\mathbf{t}_m^{\mathrm{T}}(\mathbf{x}_i - \mathbf{x}_j) + c_m)$ are positive edge weights in the patch normalized to sum to one over m, which leads to translation invariance of the weights in the feature space. The characteristics of translation invariance have better training effects when using the original spatial 3D coordinates as the input features of the shape mesh. $\mathbf{b}, \mathbf{W}_m, \mathbf{t}_m$ and c_m are trainable weights, and M is a fixed design parameter.

To achieve multi-scale convolution on meshes, we also use mesh sampling to get a new topology and connection relationship for the mesh, which helps our network capture global and local features. Specifically, we down-sample a mesh with n vertices to k vertices $(n > k)$ using permutation matrix $P_d \in \{0, 1\}^{k\times n}$ [10]. $P_d(p, q)$ denotes whether the q-th vertex is kept during down-sampling, $P_d(p, q) = 1$ if the vertex is kept, and 0 if it is discarded. The down-sampled vertex set $\mathbf{V}_d$ is a subset of the original mesh vertices. Down-sampling is obtained by iteratively contracting vertex pairs, which uses a quadratic matrix [17] to maintain surface error approximations. Up-sampling on the other hand maps k vertices to n vertices, and the up-sampling matrix $P_u \in \mathbb{R}^{n\times k}$ is constructed during down-sampling. The up-sampled vertices $\mathbf{V}_u = P_u \mathbf{V}_d$. The process of up-sampling is to re-add the vertices v_q discarded during the down-sampling process

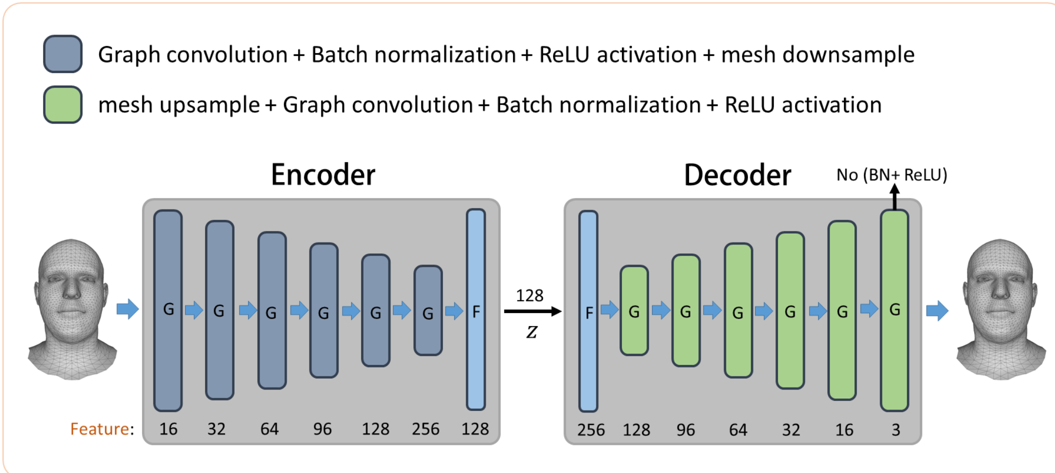

Fig. 1. The architecture of our network.

into the down-sampled mesh, *i.e.* mapping v_q into the closest triangle (h, i, j) in the down-sampled mesh and computing the barycentric coordinates by $\widetilde{v} = w_h v_h + w_i v_i + w_j v_j$ where $v_h, v_i, v_j \in \mathbf{V}_d$ and $w_h + w_i + w_j = 1$. The weights in P_u are then updated as $P_u(q, h) = w_h$, $P_u(q, i) = w_i$, $P_u(q, j) = w_j$, and $P_u(q, l) = 0$ otherwise.

In our multi-scale mesh AE, we use the per-vertex Euclidean distance between the predicted mesh and the ground-truth mesh to represent the reconstruction error because we find that better convergence can be obtained when using per-vertex Euclidean distance loss in our problem. The loss function we used is defined as

$$loss = \|D(E(\mathcal{F})) - \mathcal{F}\|^2, \tag{2}$$

The goal of our loss function is to make the reconstructed $\mathcal{F}'$ as consistent as possible with the input $\mathcal{F}$.

3.2. Network Architecture

Figure 1 shows a schematic rendition of our model. The encoder consists of 6 graph convolutions with filter dimensions of (16, 32, 64, 96, 128, 256). Each convolution is followed by a batch normalization layer [18] and a ReLU activation layer [19]. Down-sampling is used after each activation function, and the ratios are [2, 2, 2, 4, 4, 4]. The output sizes of encoder layers after graph convolution are 2512×16, 1256×32, 628×64, 157×96, 40×128, and 10×256, respectively. The last layer of encoder is a fully connected layer, which maps the feature $\in \mathbb{R}^{10\times256}$ into latent space $z \in \mathbb{R}^{128}$.

The decoder first uses a fully connected layer which maps $\mathbf{z}$ to mesh space so that we can up-sample to reconstruct the mesh. Following the fully connected layer are 6 graph convolutional layers with interleaved up-sampling layers. Each of the graph convolutions is followed by a batch normalization layer and a ReLU layer similar to the encoder network. Up-sampling ratios are [4, 4, 4, 2, 2, 2]. The output sizes of decoder layers after graph convolution are 40×128, 157×96, 628×64, 1256×32, 2512×16, and 5023×3, respectively. The last graph convolution of decoder has no batch normalization or ReLU activation.

3.3. Training Details

Our model is trained on a public face dataset, CoMA [10], which consists of 12 classes of extreme and asymmetric expressions from 12 different subjects. The dataset contains $20,466$ 3D face meshes with $5,023$ vertices and the same connectivity.

When training an auto-encoder model, we use $M = 16$ and set the latent dimension as 128, which can effectively improve the reconstruction quality. We train our multi-scale AE model directly on the input meshes with 5023 vertices, and use the 1-ring neighbors around a vertex to form the adjacency matrix. We train our auto-encoder for 100 epochs with a learning rate of 2×10^{-3} and batch size of 8 using the ADAM optimizer [20].

4. EXPERIMENTAL RESULTS

In this section, we first evaluate the reconstruction capability of our proposed model with interpolation experiment and extrapolation experiment in Sec. 4.1, and then we demonstrate the generative capacity of our model by sampling from the latent space to synthesize new expressive faces in Sec. 4.2.

4.1. Representation Quality

Interpolation Experiment. In order to evaluate the face reconstruction capability of the proposed method, we compare our method with a state-of-the-art method, CoMA [10], on its

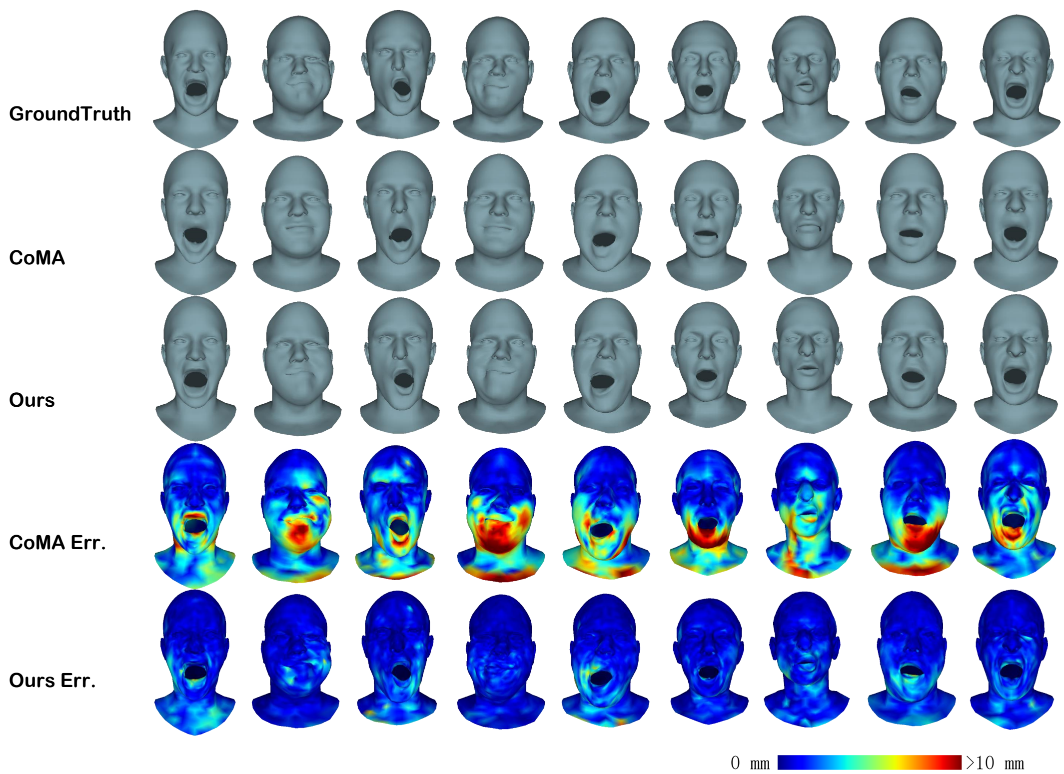

Fig. 2. Comparison with CoMA: Qualitative results for interpolation experiment.

dataset in Table 1. We adjust their network layers and parameters to best optimize their performances. The face dataset is divided into a training set and a test set with a ratio of $9:1$. The mean errors and standard deviation for per-vertex Euclidean distance on the test set are given in Table 1. It can be seen that our method achieves more accurate reconstruction. Visual inspection of the qualitative results in Figure 2 shows that our reconstructed meshes are more realistic and reasonable.

Table 1. Quantitative comparison for interpolation experiment (mm).

Sequence	Our Method		CoMA [10]	
	Mean Error	Std	Mean Error	Std
Test data	**0.583**	**0.436**	0.891	1.073

Extrapolation Experiment. To evaluate the generalization ability of our model, we verify on a cross validation dataset that splits the CoMA dataset [10] by completely excluding one expression set from all the subjects in the dataset. We test our method and the CoMA method [10] on the excluded expression. We perform 12 fold cross validation, one for each expression as shown in Table 2. It can be seen that our model performs better than the state-of-the-art method on all the expression sequences. Figure 3 shows visual inspection of the qualitative results.

Table 2. Quantitative evaluation for mesh extrapolation (mm).

Sequence	Our Method		CoMA [10]	
	Mean Error	Std	Mean Error	Std
bareteeth	**1.020**	**0.928**	1.376	1.536
cheeks in	**1.028**	**0.957**	1.288	1.501
eyebrow	**0.794**	**0.630**	1.053	1.088
high smile	**1.014**	**0.896**	1.205	1.252
lips back	**1.013**	**0.881**	1.193	1.476
lips up	**0.974**	**0.876**	1.081	1.192
mouth down	**0.803**	**0.686**	1.050	1.183
mouth extreme	**1.189**	**1.520**	1.336	1.820
mouth middle	**0.854**	**0.740**	1.017	1.192
mouth open	**0.904**	**0.815**	0.961	1.127
mouth side	**1.110**	**1.286**	1.264	1.611
mouth up	**1.005**	**0.868**	1.097	1.212

4.2. Generation of Novel Shapes

To prove the generative capabilities of our multi-scale mesh AE model, we train our model with a variational autoencoder. In our multi-scale mesh VAE, we use the per-vertex Euclidean distance to represent the reconstruction error between the predicted mesh and the ground-truth mesh. The total loss func-

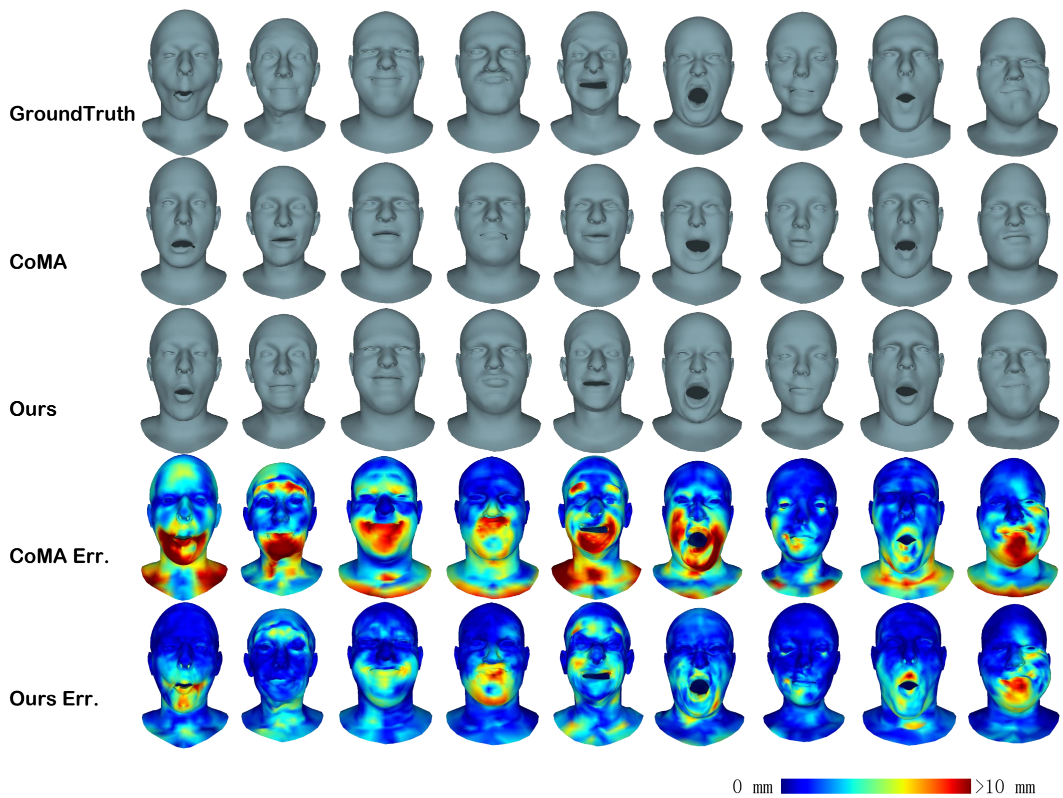

Fig. 3. Comparison with CoMA: Qualitative results for extrapolation experiment.

tion is defined as

$$loss = \|D(E(\mathcal{F})) - \mathcal{F}\|^2 + \omega D_{KL}(q(\mathbf{z}|\mathcal{F})\|p(\mathbf{z})), \quad (3)$$

where $p(\mathbf{z})$ is the prior probability, $q(\mathbf{z}|\mathcal{F})$ is the posterior probability, and D_{KL} is Kullback-Leibler divergence. The weight ω adjusts the importance of KL divergence in the latent space. Our mesh VAE is a generation model whose goal is to obtain the distribution of $q(\mathbf{z}|\mathcal{F})$, *i.e.* to obtain the distribution of the latent variable $\mathbf{z}$ given the distribution of the input data $\mathcal{F}$. It calculates $\mathcal{F}$ and joint probability distribution of $p(\mathcal{F}, \mathbf{z})$. We assume that the potential variable $\mathbf{z}$ of the output obeys a prior distribution, namely Gaussian distribution. Therefore, after training the model, we can get $\mathbf{z}'$ by sampling this prior distribution which may have not appeared in the training process but we can still get $\mathcal{F}'$ through this $\mathbf{z}'$ in the decoder. This results in new samples that match the original data distribution, and hence has the ability to generate new samples. We feed the decoder with Gaussian distribution $z \sim (0, I)$ to generate new shapes, and the results are shown in Figure 4. It can be seen that our model can easily generate plausible high-quality facial meshes. These results indicate that our self-growth weight training method can effectively balance the relationship between the variational latent distribution towards Gaussian prior and the mesh reconstruction quality. Quantitative comparison with CoMA VAE model [10] is given in Table 3. We adjust their network layers and parameters for their best performance. Our method also has smaller error for VAE model.

Table 3. Quantitative comparison for VAE experiment (mm).

Sequence	Our VAE model		CoMA VAE model [10]	
	Mean Error	Std	Mean Error	Std
Test data	**0.707**	**0.585**	1.150	1.297

5. CONCLUSION

This paper proposes a multi-scale graph convolutional model for 3D representation and reconstruction of human faces. The graph convolution algorithm based on graph structure can effectively learn mesh data, and the multi-scale sampling can make the network better learn global and local face mesh features. Experimental results demonstrate that our method produces better qualitative results and lower reconstruction errors compared with state-of-the-art method. Recovering a complete facial mesh from a poor quality depth map using our model is practically useful and will be explored in future work.

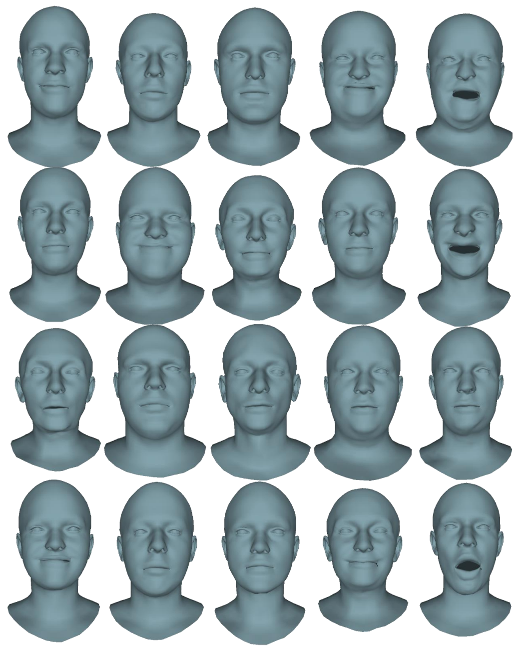

Fig. 4. Randomly generated face meshes by our multi-scale mesh VAE.

6. REFERENCES

[1] Richard A. Newcombe *et al.*, "KinectFusion: Real-time dense surface mapping and tracking," in *Proc. IEEE International Symposium on Mixed and Augmented Reality*, 2011, pp. 127–136.

[2] Volker Blanz and Thomas Vetter, "A morphable model for the synthesis of 3D faces," in *ACM SIGGRAPH*, 1999, pp. 187–194.

[3] James Booth, Anastasios Roussos, Stefanos Zafeiriou, Allan Ponniah, and David Dunaway, "A 3D morphable model learnt from $10,000$ faces," in *CVPR*, 2016, pp. 5543–5552.

[4] Tianye Li, Timo Bolkart, Michael J Black, Hao Li, and Javier Romero, "Learning a model of facial shape and expression from 4D scans," *ACM Trans. Graphics*, vol. 36, no. 6, pp. 194, 2017.

[5] John P Lewis, Ken Anjyo, Taehyun Rhee, Mengjie Zhang, Frederic H Pighin, and Zhigang Deng, "Practice and theory of blendshape facial models," in *Eurographics (State of the Art Reports)*, 2014, pp. 199–218.

[6] Pei-Lun Hsieh, Chongyang Ma, Jihun Yu, and Hao Li, "Unconstrained realtime facial performance capture," in *CVPR*, 2015, pp. 1675–1683.

[7] Justus Thies, Michael Zollhöfer, Matthias Nießner, Levi Valgaerts, Marc Stamminger, and Christian Theobalt, "Real-time expression transfer for facial reenactment," *ACM Trans. Graphics*, vol. 34, no. 6, pp. 183–1, 2015.

[8] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao, "3D ShapeNets: A deep representation for volumetric shapes," in *CVPR*, 2015, pp. 1912–1920.

[9] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas, "PointNet: Deep learning on point sets for 3d classification and segmentation," in *CVPR*, 2017.

[10] Anurag Ranjan, Timo Bolkart, Soubhik Sanyal, and Michael J Black, "Generating 3D faces using convolutional mesh autoencoders," in *ECCV*. Springer, 2018, pp. 725–741.

[11] Nitika Verma, Edmond Boyer, and Jakob Verbeek, "FeaStNet: Feature-steered graph convolutions for 3D shape analysis," in *CVPR*, 2018.

[12] Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter, "A 3D face model for pose and illumination invariant face recognition," in *CAVSBS*. Ieee, 2009, pp. 296–301.

[13] Haibin Huang, Evangelos Kalogerakis, and Benjamin Marlin, "Analysis and synthesis of 3D shape families via deep-learned generative models of surfaces," *Computer Graphics Forum*, vol. 34, no. 5, pp. 25–38, 2015.

[14] Tianfan Xue, Jianzhuang Liu, and Xiaoou Tang, "Example-based 3D object reconstruction from line drawings," in *CVPR*. IEEE, 2012, pp. 302–309.

[15] Qingyang Tan, Lin Gao, Yu-Kun Lai, and Shihong Xia, "Variational autoencoders for deforming 3D mesh models," in *CVPR*, 2018.

[16] Lin Gao, Jie Yang, Yi-Ling Qiao, Yu-Kun Lai, Paul Rosin, Weiwei Xu, and Shihong Xia, "Automatic unpaired shape deformation transfer," *ACM Trans. Graphics*, vol. 37, no. 6, pp. 1–15, 2018.

[17] Michael Garland and Paul S Heckbert, "Surface simplification using quadric error metrics," in *ACM SIGGRAPH*, 1997, pp. 209–216.

[18] Sergey Ioffe and Christian Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *ICML*, 2015, pp. 448–456.

[19] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng, "Rectifier nonlinearities improve neural network acoustic models," in *ICML*, 2013, p. 3.

[20] John Duchi, Elad Hazan, and Yoram Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, no. Jul, pp. 2121–2159, 2011.