Plausible 3D Face Wrinkle Generation Using Variational Autoencoders

Qixin Deng, Luming Ma, Aobo Jin, Huikun Bi, Binh Huy Le, and Zhigang Deng, *Senior Member, IEEE*

Abstract—Realistic 3D facial modeling and animation have been increasingly used in many graphics, animation, and virtual reality applications. However, generating realistic fine-scale wrinkles on 3D faces, in particular, on animated 3D faces, is still a challenging problem that is far away from being resolved. In this article we propose an end-to-end system to automatically augment coarse-scale 3D faces with synthesized fine-scale geometric wrinkles. By formulating the wrinkle generation problem as a supervised generation task, we implicitly model the continuous space of face wrinkles via a compact generative model, such that plausible face wrinkles can be generated through effective sampling and interpolation in the space. We also introduce a complete pipeline to transfer the synthesized wrinkles between faces with different shapes and topologies. Through many experiments, we demonstrate our method can robustly synthesize plausible fine-scale wrinkles on a variety of coarse-scale 3D faces with different shapes and expressions.

Index Terms—Face modeling, wrinkle synthesis, deep generative models, variational autoencoders

1 Introduction

Realistic three-Dimensional (3D) facial animation has attracted many attentions in both academia and industry in recent decades [1], [2], [3], due to the emergence of its versatile applications in virtual reality, films, games, education, training, and so on. Despite numerous progresses made, creating 3D face models with rich details is still non-trivial, which requires not only coarse-scale facial features but also subtle fine-scale facial features like wrinkles. Such geometric details often convey important characteristics, and significantly affect the visual appearance of the face. But geometric details like wrinkles have been often under-explored in many popular face modeling methods [4], [5], [6]. Indeed, manually crafting wrinkles using 3D modeling packages is often non-trivial, cumbersome, and labor-intensive; this task could become even almost infeasible when dealing with animated faces.

High-end face performance capture techniques are able to reconstruct high quality facial details but significantly rely on structured light and photo-metric stereo for face scanning [3], which is often limited in a controlled environment and/or involves non-trivial intrusive setups. Recent techniques can extract such geometric details from monocular video [3], [7], [8], [9], [10], but they require substantial computational time, and are generally sensitive to face poses and lighting conditions of the input video. More importantly, the above methods, including high-end facial performance capture and monocular video based methods, cannot be used to add geometric wrinkles to novel 3D face models, which substantially limits their practical applications in movie/game industries. To tackle this issue, researchers started to explore flexible alternatives such as sketch based wrinkle generation [11]. However, it is non-trivial for users, in particular, novices, to manually draw plausible wrinkles on 3D face models using such methods. Furthermore, manually preserving the consistency of the drawn wrinkles across animated facial frames could quickly become overwhelming or maybe even an impossible task.

Clearly, there are several challenges standing in front of an ideal realistic 3D wrinkle generation method: (1) It should not rely on any prior knowledge of wrinkles for a coarse-scale target face model, that is, wrinkles of the target coarse-scale face model in other expressions may be unobserved but need to be plausibly predicted. (2) It should be sufficiently generalized to handle a variety of 3D face models with different identities and expressions. This is non-trivial since the shape and topology of target face models might be dramatically different from a standard face model. (3) It should be capable of preserving the wrinkle consistency during animation. In other words, animated wrinkles are expected to be spatially and temporally consistent.

Inspired by the above challenges, in this paper we propose a novel 3D face wrinkle synthesis approach using a deep generative model, and also introduce a complete wrinkle transfer pipeline that can be robustly applied to a variety of 3D face models with different identities and expressions. Specifically, to obtain the training data, we first employ a recent monocular video based capture method [10] to acquire geometric wrinkles as displacements along per-vertex normals. Then, by formulating the wrinkle generation problem as a supervised generation task, we implicitly model the continuous space of

- *Qixin Deng, Luming Ma, Aobo Jin, and Zhigang Deng are with the Department of Computer Science, University of Houston, Houston, TX 77204 USA. E-mail: {qxdeng1991, balokume, jinaobo1103}@gmail.com, zdeng4@uh.edu.*
- *Huikun Bi is with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100864, China. E-mail: xiaobi361@gmail.com.*
- *Binh Huy Le is with the SEED Lab, Search for Extraordinary Experience Division, Electronic Arts Inc., Ashburn, VA 20147 USA. E-mail: bbinh85@gmail.com.*

Manuscript received 22 Apr. 2020; revised 18 Nov. 2020; accepted 10 Jan. 2021. Date of publication 13 Jan. 2021; date of current version 1 Aug. 2022. (Corresponding author: Zhigang Deng.) Recommended for acceptance by N. J. Mitra. Digital Object Identifier no. 10.1109/TVCG.2021.3051251

face wrinkles via a compact generative model such that plausible face wrinkles can be generated through effective sampling and interpolation in the space. In addition, in order to generate wrinkles on a variety of novel target faces, we also design a complete pipeline to faithfully transfer the wrinkles from the reference face to various target face models.

The main contributions of our work can be summarized below:

- We introduce the first deep generative model for directly synthesizing 3D geometric wrinkles on novel target coarse-scale 3D faces. It also can handle wrinkle generation on animated 3D faces, while preserving the spatial and temporal consistencies of the wrinkles across frames.
- We introduce a complete pipeline to faithfully transfer geometric wrinkles from the reference face to novel target faces. In this way, our approach can be used to generate geometric wrinkles on 3D faces with different shapes and expressions.

2 Related Work

In this section, we briefly review recent research efforts that are most related to this work, including facial performance capture, variational autoencoders, and wrinkle synthesis. For comprehensive surveys on facial animation and modeling, readers are referred to recent relevant surveys [1], [2], [3].

Face Performance Capture. Researchers employ structured light and photometric stereo [12] or attach markers on the face [13], [14] to acquire the dynamic deformation of 3D facial performance. The above methods are able to obtain 3D facial models with geometric details, but inevitably involve complex intrusive setups for subjects. Later, researchers exploited stereo images [15], [16], [17], [18] or light-weight binocular cameras [19], but these methods are limited to the requirement of binocular video footages or depth data. In recent years direct reconstruction from monocular video [7], [9], [20], [21] becomes increasingly popular due to its low cost setup and compatibility with various legacy video footages. Follow-up works [8], [22] build controllable face rigs and appearance from video for animation. All the above methods, however, require intensive off-line processing.

To meet the demand of real-time applications, real-time facial tracking systems have been developed using structured light [23]. Such methods offline fit a reference face model for an individual first and then do online tracking for expression transfer. Many methods have also been proposed to combine depth information from a single RGB-D camera [24], [25], [26] to track facial deformations in real-time. Based on regression algorithms, Cao *et al.* [27] proposed a real time face tracking method to capture coarse 3D facial geometry from a single monocular camera. Their follow up work [28] learns displacement patches from captured texture to predict medium-scale geometry details. Recently, many deep learning methods have been proposed to reconstruct facial performance from input images or video [29], [30]. To achieve fine-scale facial geometric details comparable to offline qualities, Ma and Deng proposed a novel hierarchical reconstruction method [10] to reconstruct high resolution facial geometry and appearance in real-time based on a single monocular RGB video clip.

Variational Autoencoders. Deep learning models have been increasingly used for computer graphics applications in recent years. For example, multi-view Convolutional Neural Networks (CNNs) [31] render 3D points clouds or meshes into depth maps, which are suitable to be processed by CNNs. Tatarchenko *et al.* [32] utilize the encoder-decoder architecture to predict multi-views of a given object in 2D space. The framework of variational autoencoders (VAE) [33] provides a principal method for jointly learning deep latent variable models and corresponding inference models using stochastic gradient descent.

Many works have been done to extend the VAE models for graphics applications. Researchers apply 3D CNN to variational autoencoders such that 3D volumetric objects are embedded into a compact space [34]. Nash and Williams [35] proposed the ShapeVAE model to generate point coordinates and normals based on different parts of an object. Li *et al.* [36] proposed the GRASS model, a generative recursive autoencoder for shape structures. Tan *et al.* [37] use mesh variational autoencoders to generate high quality deformable models with rich details. Ranjan *et al.* [38] proposed a mesh autoencoder which is able to produce deformed face meshes via sampling and interpolation in the latent space. Recently, Saito *et al.* [39] proposed a fully automatic single-view 3D hair reconstruction method based on the VAE framework. Gao *et al.* [40] generate 3D shapes as structured deformable meshes based on the VAE framework. Similarly, Mo *et al.* [41] proposed the StructureNet to learn a generative autoencoder of shape structures based on graph neural networks.

In this work, our generative model predicts plausible face wrinkles using variational autoencoders with convolutional layers. Because CNNs are originally designed to process images, researchers often have to parameterize their problems into regular grid representations for graphics applications. As such, we convert wrinkles into a tensor, which is suitable to be processed by CNNs.

Wrinkle Synthesis. Face wrinkles are strongly related to physical muscles of the face. Hence, several methods have been proposed to generate wrinkles under specific face deformations by using simplified physical models of facial structures. Zhang *et al.* [42] proposed a mass-spring system to simulate the elastic dynamics of expressive facial wrinkles. Their follow-up work [43], [44] describes a subsequent multi-layer deformation model to synthesize facial expressions. Some researchers aimed at accurate reconstruction of age-related skin wrinkles [45], [46]. All the above methods are typically computationally complex, and heavily rely on non-trivial tuning of physical parameters.

In recent years, geometric modeling methods have attracted increasing attentions for wrinkle synthesis. Researchers proposed a curvature driven model using controllable energies to control various wrinkle shapes [47], [48]. Cutler *et al.* [49] proposed a method to generate wrinkles on animated characters, but users have to manually place a set of shape points on the character. However, with the above methods, it is still less intuitive for users to estimate the parameters to achieve realistic facial wrinkles for a specific expression, besides involved with painstaking manual efforts. In light of

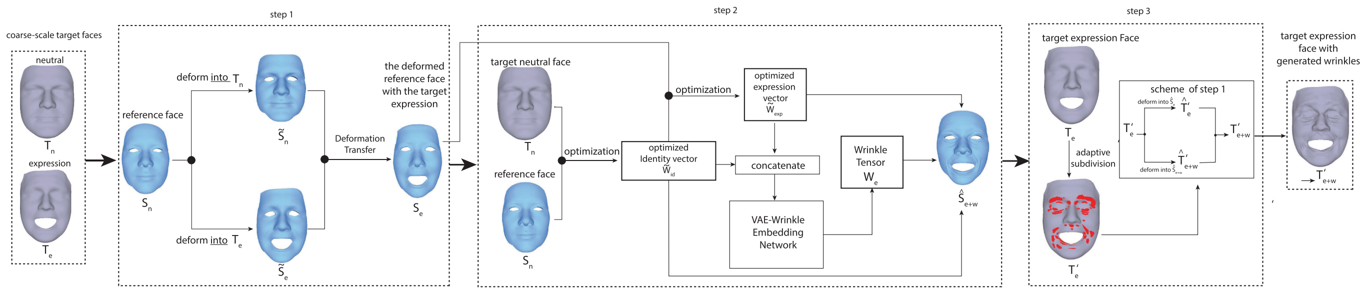


Fig. 1. Pipeline illustration of our wrinkle synthesis and transfer pipeline. It consists of three main steps: (1) deform the reference face with the target face expression to obtain S_e, (2) synthesize the wrinkles on the deformed reference face with the optimized identity and expression vectors, $\hat{S}_{e+w}$, and (3) transfer the wrinkles from $\hat{s}_{e+w}$ to the subdivided target expression face model $T_e^{'}$, and obtain the final target expression face with the transferred wrinkles, $T_{e+w}^{'}$.

the above issue, Kim *et al.* [11] proposed a sketch based wrinkle generation method, which gives users capability to directly design and control the amount and shapes of wrinkles on 3D face models. But drawing well-placed sketch curves on 3D meshes is non-trivial, and it becomes even more difficult for users to draw consistent wrinkles for different expressions or for animations.

It is noteworthy that the method by Ma *et al.* [50] can also add wrinkles to 3D face models based on polynomial displacement maps, but there are several significant differences between it and our work: (1) Except the captured subject's face, their method [50] cannot be generalized or applied to any other face models, without acquiring new data or training new models. In contrast, our work allows users to straightforwardly apply our trained model to other face models without extra data acquisition efforts. (2) Their method [50] needs to capture fine-scale geometry/texture and motion capture data of the subjects simultaneously as training data, while our method does not need simultaneously acquired facial mocap data. Because of the above significant differences, it is difficult to perform a fair comparison between our work and the method in [50].

3 Approach Overview

Our wrinkle synthesis approach can be conceptually decomposed into the following two main modules: (i) VAE-Wrinkle Embedding Network (VAE-WEN) construction, and (2) wrinkle synthesis and transfer, described below.

VAE-WEN Construction. First, based on an in-house collected 3D facial wrinkles dataset (Section 4), which is directly extracted from high quality monocular face video [10], we transform the wrinkles data into tensors (called *wrinkle tensors*), which are suitable to be processed by CNNs. Then, we build a deep learning based model to transform the wrinkle tensors into compact latent representations, and thus plausible new wrinkles can be sampled and interpolated in the latent space (called *VAE-WEN*) (Section 5).

Wrinkle Synthesis and Transfer. Given a coarse-scale target expression face (without wrinkle details) as well as its corresponding coarse-scale neutral face as the input, the wrinkle synthesis and transfer module (Section 6) is designed to first synthesize wrinkles on the reference face and then faithfully transfer the wrinkles to the target expression face. Technically, it can be further decomposed into the following three steps. Fig. 1 illustrates the pipeline of the wrinkle synthesis and transfer module.

- *Step 1 (deform the reference face with the target expression)*: Via a registration process, we first deform the reference face, S_n, to resemble the target neutral face, T_n, and the target expression face, T_e, as much as possible, and then we further extend the deformation transfer algorithm [51] to deform the reference face to have the same expression as the target expression face, and obtain S_e.
- *Step 2 (wrinkle synthesis)*: Through the constructed bilinear face model, where each face can be compactly characterized as the combination of an identity vector and an expression vector, we employ a two-steps optimization process to generate the optimized identity vector, w_{id}, and the optimized expression vector, w_{ex}. Then, taking both w_{id} and w_{exp} as the input, our constructed VAE-WEN can synthesize fine-scale wrinkles on the deformed reference face with the optimized target expression. We denote the resulting face as $\hat{S}_{e+w}$.
- *Step 3 (wrinkle transfer)*: Through adaptive subdivision and a similar transfer scheme as in the above step 1, we can transfer the fine-scale wrinkles from $\hat{S}_{e+w}$ (resulted from the step 2) to the subdivided target expression face, $T_e^{'}$. The final output of step 3 is a subdivided target expression face with synthesized fine-scale wrinkles, denoted as $T_{e+w}^{'}$.

4 Wrinkle Data Acquisition

To train our model, we need to collect a large set of 3D geometric wrinkles on faces with various shapes and expressions. In this work we employ one of recent facial performance capture methods [10] to reconstruct 3D geometric wrinkles directly from high resolution monocular video.

Specifically, we used publicly available face video datasets ([52] and [53]) and down-sampled all the videos to 10 fps. We obtained a total of 30,457 video frames of 50 individuals with various expressions. We split the data into a training set (27,446 frames) and a test set (3,011 frames). Given a monocular front face video, we first fit the FaceWareHouse bilinear face model [4] to the input video by estimating camera parameters, an identity vector, an expression vector, and head pose. The FaceWareHouse bilinear face model is created by using facial geometry of 150 subjects with 47 facial expressions. All data can be assembled into a rank-3 data tensor T (11K vertices$\times$150 identities $\times$ 47 expressions). Then we use N-mode singular value decomposition (SVD) to decompose the tensor. The N-mode SVD process is represented as

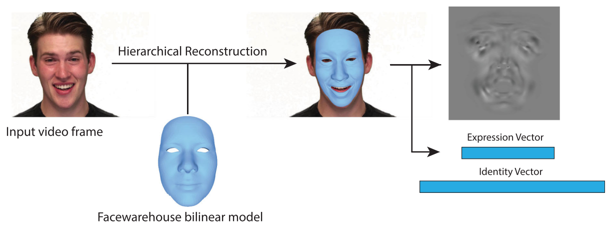


Fig. 2. 3D geometric wrinkle data acquisition process in this work.

$$T \times_2 U_{id}^T \times_3 U_{exp}^T = C, \tag{1}$$

where T is the data tensor, and C is called the core tensor. The N-mode SVD helps to "rotate" the data tensor and sort the variances of C in a decreasing order for each mode. This allows us to truncate the insignificant components of C and obtain a reduced model of the dataset to approximate the original data tensor as

$$T \simeq C_r \times_2 \hat{U}_{id} \times_3 \hat{U}_{exp}, \tag{2}$$

where C_r is the reduced core tensor produced by retaining the top-left corner of the original core tensor, $\hat{U}_{id}$ and $\hat{U}_{exp}$ are the truncated matrices from U_{id} and U_{exp} by removing the trailing columns. We call C_r the bilinear face model from the FaceWareHouse dataset. With C_r, any facial expression of a subject can be approximated by tensor contraction

$$V = C_r \times_2 w_{id}^T \times_3 w_{exp}^T, \tag{3}$$

where w_{id}^T and w_{exp}^T are the identity vector (50 $\times$ 1) and the expression vector (25 $\times$ 1), respectively. To this end, we obtained one identity vector for each individual and one expression vector for each video frame. Then, by employing the hierarchical reconstruction and displacements refinement algorithms [10], we can augment the initially reconstructed, coarse-scale face mesh with geometric wrinkle details. Finally, we stored the extracted wrinkles as 1024 $\times$ 1024 *wrinkle tensors*. A 2D wrinkle tensor in this work is defined as per-vertex displacements along vertex normals, which are stored in a 2D format using the UV coordinates of the vertices, and the gaps between non-zero elements are bilinearly interpolated. Meanwhile, we also stored the *identity vector* (i.e., identity weights) and the *expression vector* (i.e., expression weights) used in the bilinear face model [4] that is associated with each stored wrinkle tensor. This wrinkle data acquisition process is illustrated in Fig. 2.

5 VAE-Wrinkle Embedding Network

In this section, we first describe our network architecture, including variational autoencoders and a wrinkle embedding network. After that, we evaluate the loss functions used in our VAE network.

5.1 Network Architecture

Variational Autoencoders. Our approach is based on the variational autoencoder (VAE), which has become one of widely used generative models in recent years. A classic VAE model consists of an encoder $E_\theta(x)$, and a decoder $D_\phi(z)$. The encoder $E_\theta(x)$ encodes an input x into a latent space as a latent vector z, while the decoder $D_\phi(z)$ can generate an output x' from a given latent vector z. Different from the vanilla autoencoder which approximates $E_\theta(x)$ by using a deterministic function, VAE uses a posterior distribution $q(z|x)$ to approximate $E_\theta(x)$ such that VAE is able to generate new data x' by sampling z from a prior distribution $p_\phi(x|z)$. We train the encoding and decoding parameters θ and ϕ using stochastic gradient variational Bayes (SGVB) algorithm [33] as follows:

$$\theta^*, \phi^* = \arg\min_{\theta,\phi} \mathbb{E}_{z\sim E_\theta(x)}[-\log p_\phi(x|z)] + \mathbb{D}_{kl}(E_\theta(x)|p(z)), \tag{4}$$

where $\mathbb{D}_{kl}$ denotes the Kullback-Leibler divergence, which measures the difference between $E_\theta(x)$ and $p(z)$. Specifically, assuming $p(z)$ is a standard Normal distribution $N(0,1)$ as a prior, and a posterior Gaussian distribution $E_\theta(x) \sim N(z_\mu, diag(z_\sigma))$, then the Kullback-Leibler divergence $\mathbb{D}_{kl}$ is formulated as

$$\mathbb{D}_{kl}(N(z_\mu, diag(z_\sigma))|N(0,1)) = \frac{1}{2}\sum_i (z_{\sigma,i}^2 + z_{\mu,i}^2 - log(z_{\sigma,i}^2) - 1). \tag{5}$$

To make all operations differentiable for back-propagation, the latent vector z is to be sampled through a reparameterization trick [33] as

$$z = z_\mu + \epsilon \odot z_\sigma, \epsilon \sim N(0,1), \tag{6}$$

where $\odot$ is an element-wise matrix multiplication operator, and z_μ and z_σ are the multi-dimensional outputs of $E_\theta(x)$.

VAE Architecture. To obtain the space of human face wrinkles, we train our VAE network using the extracted wrinkle tensors (Section 4). Specifically, we reshape each wrinkle tensor as $1024 \times 1024 \times 1$ before feeding it into the encoder. The encoder is to encode the input wrinkle tensors into a latent space z_μ and z_σ with 128 dimensional parameters. Then, we sample z from z_μ and z_σ using the re-parameterization trick introduced in [33]. The latent vector can be used as the input of the decoder and reverted to a $1024 \times 1024 \times 1$ tensor. Fig. 3 illustrates our network architecture.

Loss Functions. We utilize two loss functions to train the network weights. The first one is the reconstruction loss denoted as $\mathcal{L}_{recon}$, which aims to minimize the reconstruction error between the input wrinkle tensor and the output of the decoder. The second one is the KL divergence loss denoted as $\mathcal{L}_{KL}$. For reconstruction, we use the *L1* loss between the elements of the input wrinkle tensor I and the decoder-generated wrinkle tensor O, because it leads to less smoothing results compared to L2, given by

$$\mathcal{L}_{recon} = \frac{1}{N}\sum_i \sum_j \|I_{i,j} - O_{i,j}\|_1, \tag{7}$$

where N denotes the total number of the elements in a wrinkle tensor. But the wrinkle details from the VAE network are typically over-smoothed. To address this issue, we apply a wrinkle edge-aware weighting scheme on the wrinkle tensor to make the generation of wrinkles more sensitive to wrinkle edges

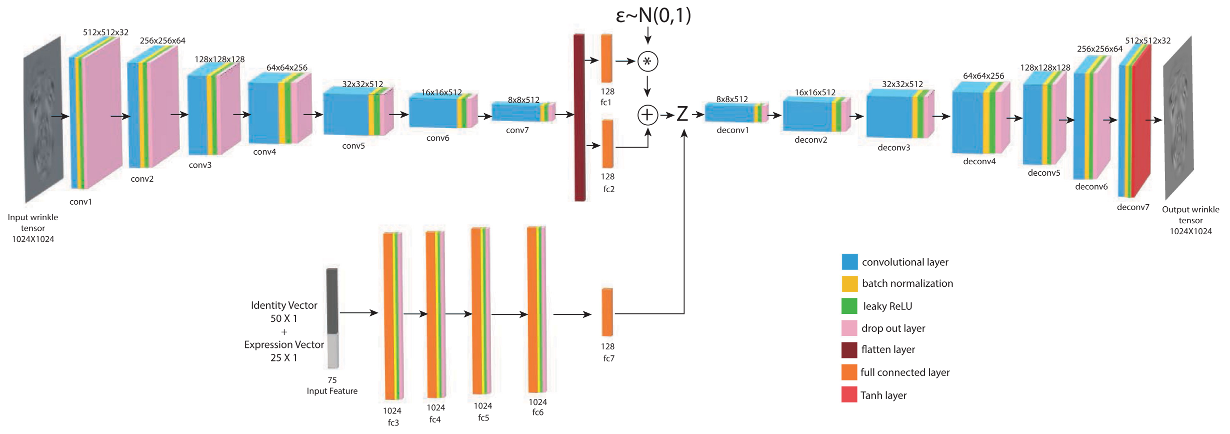


Fig. 3. Our network architecture overview. Our VAE consists of an encoder and a decoder, the encoder takes wrinkle tensors as inputs, and the decoder is able to generate wrinkle tensors. Our wrinkle embedding network aims to sample the latent space z, therefore, the decoder of the VAE can produce corresponding wrinkle tensors.

$$\begin{aligned}\mathcal{L}_{recon} =& \frac{1}{N}\sum_i\sum_j w_{i,j}\|I_{i,j} - O_{i,j}\|_1,\\ w_{i,j} =& \log(\alpha G_{i,j} + 1.0).\end{aligned} \tag{8}$$

For an input wrinkle tensor I, we have G_{row} as the derivative of I with respect to the row direction and G_{col} as the derivative of I with respect to the column direction. After that, we set each element of G, $G(i,j) = \max(G_{row}(i,j), G_{col}(i,j))$. The log function in Eq. (8) aims to balance the weights between large and small wrinkles. α is a constant, which scales G into a proper range. In our experiments we set $\alpha = 1000$.

Our final loss function is defined as follows:

$$\mathcal{L}_{VAE} = w_{recon}\mathcal{L}_{recon} + \mathcal{L}_{KL}. \tag{9}$$

The VAE encoder and decoder networks have similar architectures, with kernel size = 5, stride = 2, and padding = 1 for all of convolution layers. After each convolution layer, we add a batch normalization layer, leaky ReLU activation function, and a dropout layer, except the last layer of the decoder, which is followed by a *Tanh* activation function. We set w_{recon} to 100, the keep probability of the dropout layer to 0.8, and β in the leaky ReLU activation to 0.2. We trained the VAE network for 50 epochs with the Adam solver. The batch size is 6, and the learning rate is set to 10^{-4}.

Wrinkle Embedding Network (WEN). The latent vectors of wrinkles form a space. To achieve wrinkle synthesis, we train a WEN to predict the wrinkle latent vector z in the space from face features. As described in Section 4, for each face, we generate its identity vector and its expression vector and then concatenate them together to form a face feature vector. The goal of the WEN is to map a given face feature vector y to the wrinkle latent space z. We use fully connected networks to build the WEN. The WEN $\mathcal{F}$ takes a face feature vector y as input and predicts the latent space vector $\hat{z}$. The loss function is used to minimize the distance between the encoder output z and the WEN's output $\hat{z}$, defined as below:

$$\mathcal{L}_{\mathcal{F}} = ||z - \hat{z}||_2. \tag{10}$$

The WEN consists of four 1024-dimensional, fully connected layers. Each layer is followed by batch normalization, leaky ReLU activation, and dropout layer. Similar to the VAE network, we set the keep probability of the dropout layer to 0.8, and 0.2 for learky ReLU activation function. We trained our WEN for 75 epochs with the Adam solver. The batch size is 12, and the learning rate is set to 10^{-3}.

5.2 Evaluation

We compared our wrinkle edge-aware reconstruction loss function (Eq. (8)) with the L1 loss function (Eq. (7)) for VAE model and WEN model. For a fair comparison, we used the same architecture and parameters, except different loss functions, for training. We tested two trained models on our test set (total 3,011 face frames, each of which has a 1024 × 1024 wrinkle tensor). We use the per-element error to evaluate each model performance. Table 1 shows the per-element euclidean errors when using the VAE and WEN models, with either our proposed wrinkle edge-aware loss function or the L1 loss function, to generate wrinkle tensors based on our test set. We can clearly see that in both the VAE and WEN models, our wrinkle edge-aware loss function can generate more accurate wrinkle tensors than the L1 loss function. Fig. 4 shows a comparison example, which demonstrates that the generated results by our wrinkle edge-aware loss function match the ground-truth data closely, whereas the L1 loss function leads to less accurate and over-smoothed results. We circle two regions (red and black) in this example, and there exist obvious visual differences between our wrinkle edge-aware loss function and the L1 loss function.

TABLE 1
Per-Element Euclidean Errors When Using Our VAE and WEN Models, With Either Our Proposed Wrinkle Edge-Aware Loss Function or the L1 Loss Function, to Generate Wrinkle Tensors From Our Test Set

Model	Mean Error ± SD	Median Error
VAE with our proposed Loss	$\mathbf{5.43 \pm 4.91}$	**3.96**
VAE with L1 Loss	7.85 ± 8.79	6.32
WEN with our proposed Loss	$\mathbf{5.81 \pm 5.10}$	**4.27**
WEN with L1 Loss	8.45 ± 9.03	6.86

The unit of the per-element errors in this table is 10^{-3}.

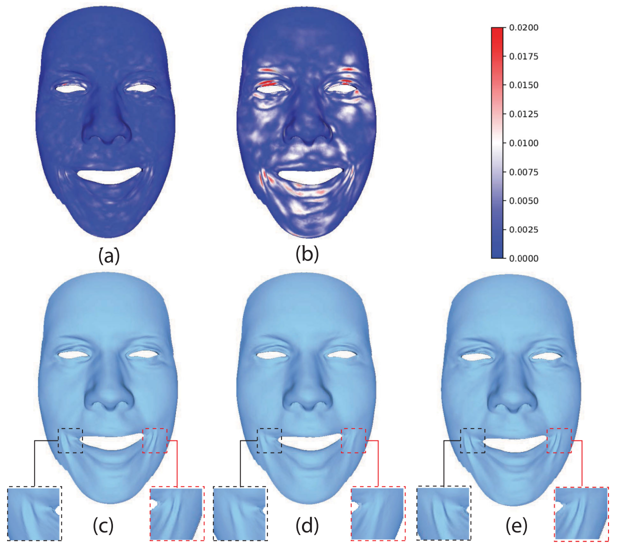


Fig. 4. Comparison of different training loss functions. (a) Is the heat map between the wrinkle generated by the deep learning model trained with our wrinkle edge aware loss function and the ground truth. (b) Is the heat map between the wrinkle generated by the deep learning model trained with L1 loss and the ground truth. (c) Is the 3D face applied with the wrinkle tensor generated by the VAE-WEN with our wrinkle edge-aware loss function. (d) Is the 3D face applied with the wrinkle tensor generated by the VAE-WEN with L1 loss function. (e) Is the 3D face applied with the ground truth wrinkle tensor.

6 Wrinkle Synthesis and Transfer

Since the number of vertices and the topology of different faces are varied, and our wrinkle tensors are generated based on the geometry of the reference face, we cannot directly apply our wrinkle tensor onto novel target face models. To this end, we design a complete pipeline for wrinkle synthesis and transfer, described below. Notations used in this section are summarized in Table 2.

6.1 Deform the Reference Face With Target Expression

In order to transfer the wrinkle on the reference face to a target face, we take inspiration from the deformation transfer method [51] that can retarget the deformation of one mesh to another, not constrained by the number of vertices and topology, assuming a few point-to-point correspondences are provided. The transfer maintains the target mesh identity and keeps a proper scaling of deformation. Therefore, before deformation transfer, we need to do mesh registration between the reference face and the target face.

Different from [51] that builds the triangle correspondences between the source and the target meshes through vertex-wise correspondences, in this work we introduce a different scheme. First, using the vertex-wise correspondences, we deform the reference neutral face S_n *into* the target neutral face T_n and the target expression face T_e, respectively, and obtain two meshes $\tilde{S}_n$ and $\tilde{S}_e$. Note that $\tilde{S}_n$ (or $\tilde{S}_e$) has the same geometric topology as S but its shape closely approximates T_n (or T_e), and thus $\tilde{S}_n$ and $\tilde{S}_e$ do not maintain the identity of S.

To obtain the deformed reference face S_e that has the expression of T_e while maintains the S identity, we solve the following minimization problem:

TABLE 2
Notations Used in Section 6

Notation	Explanation
S_n	the reference neutral face
$\tilde{S}_n$	S_n deformed into the target neutral face
$\tilde{S}_e$	S_n deformed into the target expression face
S_e	the deformed reference face with the target expression
$\hat{S}_e$	reference face constructed based on optimized feature vectors (identity and expression vectors))
$\hat{S}_{e+w}$	$\hat{S}_e$ applied with the generated wrinkles
T_n	the target neutral face
T_e	the target expression face
$T_e^{'}$	the adaptively subdivided version of T_e
$\hat{T}_e^{'}$	$T_e^{'}$ deformed into $\hat{S}_e$
$\hat{T}_{e+w}^{'}$	$T_e^{'}$ deformed into $\hat{S}_{e+w}$
$T_{e+w}^{'}$	target expression face with the generated wrinkles

$$\arg\min_{\hat{v}_1 \ldots \hat{v}_n} \sum_i \|M_{s_i} - M_{t_i}\|^2, \tag{11}$$

where M_{s_i} is the affine transformation matrices for all triangles between $\tilde{S}_n$ and $\tilde{S}_e$, M_{t_i} is the affine transformation matrices between the reference neutral face S_n and the to-be-solved reference expression face S_e (with the new vertex positions $\hat{v}_1 \ldots \hat{v}_n$).

We also compared our transfer scheme with the original scheme in [51]. For a fair comparison, we used the same input face models and markers. Fig. 5 shows a comparison example of two different transfer schemes. Its left side uses a target face [5] that has 3,448 vertices and 6,736 triangles, and its right side uses a target face [54] that has 28,588 vertices and 56,572 triangles. The comparisons in Fig. 5 clearly show that our method can outperform the method in [51] for both of the face meshes with a total of six expressions. In particular, the deformation transfer method in [51] fails to preserve the expressions on the right target face that has many more vertices and triangles than the FaceWareHouse reference face model (5,639 vertices and 10,988 triangles). The reason is that, since the deformation transfer method [51] computes the triangle face correspondences between the source and target meshes, in the case of the right target face in Fig. 5, a target triangle may correspond to multiple source triangles, which could lead to the lack of necessary constraints in linear systems. In contrast, our method builds the linear systems to be one-to-one correspondences between source triangles and target triangles. This ensures that our systems have stronger constraints and thus produce more accurate deformations.

To validate the necessity of deformation transfer in our framework, we did a comparison experiment, where we directly used the non-rigid registration result $\tilde{S}_e$ as the input of step 2. The comparison results are shown in Fig. 6. From Fig. 6, we can clearly observe that our framework with the use of deformation transfer obtains much better results than our framework without the use of deformation transfer (i.e., directly use non-rigid registration results). The main reason is that, our training data are generated by using the FaceWareHouse face model; if we directly use non-rigid registration results, the identity of the deformed source mesh cannot be preserved, which could lead to poor results in the WEN

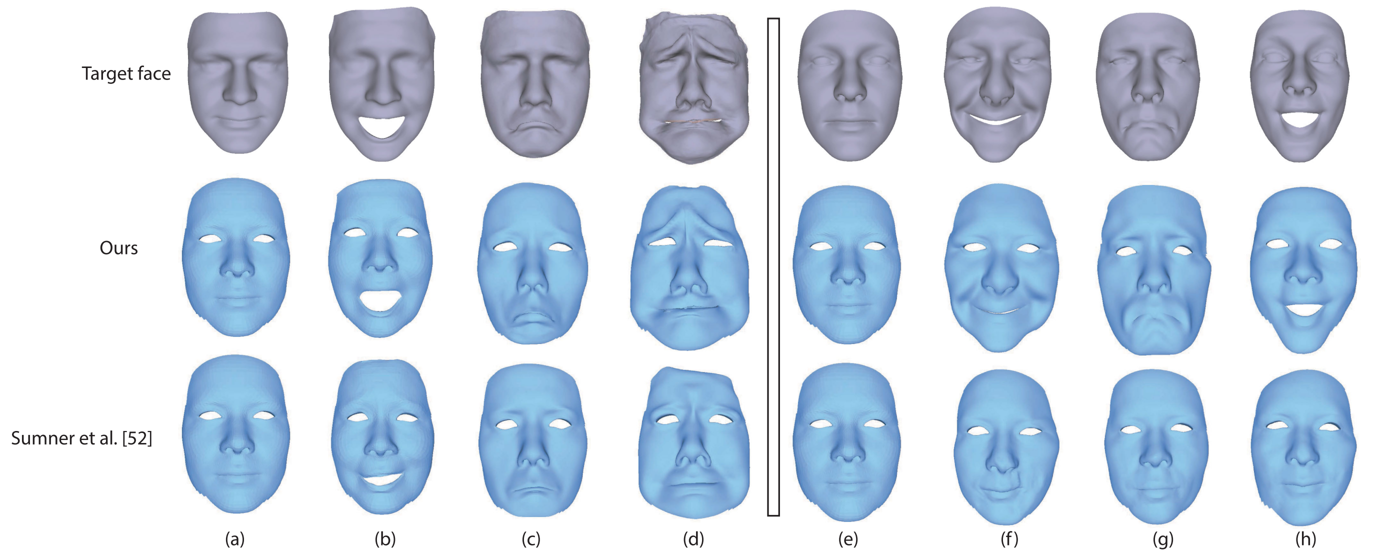


Fig. 5. Comparisons of two deformation transfer schemes. The extended deformed face scheme in our approach can generate more accurate facial expression deformations than the original deformation transfer algorithm [51], in particular, if the target face mesh has many more vertices/triangles than the reference face model (i.e., the right example).

model. Therefore, it falls short of generating target face meshes with desired wrinkle details.

6.2 Wrinkle Synthesis

To generate wrinkles on the target expression face model via our VAE-WEN, we need to first extract the identity vector and the expression vector from the target expression face. As we do not have any ground-truth wrinkles for the target expression face (only have the 3D coarse-scale target expression face), we cannot directly obtain the two vectors using the method in [10].

To address this problem, we first solve the following optimization problem to obtain the optimized identity vector $\tilde{w}_{id}$

$$\arg\min_{w_{id}} \sum_i \|C_{r,i} \times_2 w_{id}^T \times_3 w_{neutral}^T - V_{marker,i}\|^2, \tag{12}$$

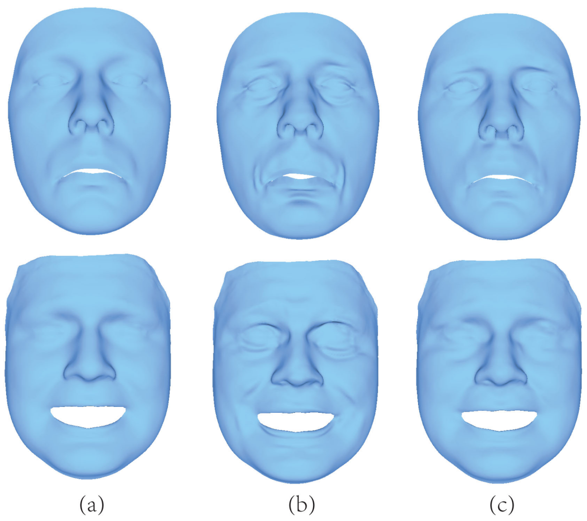


Fig. 6. Result comparisons with/without the deformation transfer step. Column (a) shows two different target expression faces. Column (b) shows the generated result by our pipeline with our proposed deformation transfer scheme. Column (c) shows the generated result by our pipeline without deformation transfer.

where C_r is the bilinear face model built from the FaceWareHouse dataset [4], $w_{neutral}$ is the expression vector of the reference neutral face S_n, and V_{marker} is the set of user-specified vertices used in deformation transfer.

After we obtain $\tilde{w}_{id}$, we can further solve the optimized expression vector $\tilde{w}_{exp}$. We solve a similar optimization problem as follows:

$$\arg\min_{w_{exp}} \sum_i \|C_{r,i} \times_2 w_{id}^T \times_3 w_{exp}^T - V_{marker,i}\|^2. \tag{13}$$

After we obtain $\tilde{w}_{id}$ and $\tilde{w}_{exp}$ for the target expression face, we just need to feed $\tilde{w}_{id}$ and $\tilde{w}_{exp}$ into our constructed VAE-WEN, and its output is the generated wrinkle tensor, $\mathbf{W}_e$, for the target expression face.

6.3 Wrinkle Transfer

Due to the shape difference between the reference face and the target expression face T_e, we cannot directly apply the above $\mathbf{W}_e$ onto T_e. In this work, we use a scheme similar to the one in Section 6.1 for wrinkle transfer, described below. Assume $\hat{S}_{e+w}$ is the face model resulted from applying $\mathbf{W}_e$ on the deformed reference face model $\hat{S}_e$ that is constructed based on the $\tilde{w}_{id}$ and $\tilde{w}_{exp}$ (obtained in Section 6.2). Different from the coarse-scale expression transfer scenario in Section 6.1, at this step we need to transfer the fine-scale wrinkles from $\hat{S}_{e+w}$ to T_e.

To make the linear systems (refer to Section 6.1) more sensitive to wrinkles, we first adaptively subdivide the target expression face T_e. The used subdivision criterion is based on the wrinkle level on each triangle face. Specifically, as described in Section 4, the wrinkle at each vertex v_i is represented as the displacement (denoted as d_{v_i}) at the vertex along its normal direction. Then, if all the three wrinkle displacements at the three vertices of a triangle (assuming v_{i1}, v_{i2}, and v_{i3}) are larger than a threshold (assuming the used face meshes are triangle meshes), that is, $\max(d_{v_{i1}}, d_{v_{i2}}, d_{v_{i3}}) > \epsilon$, we then iteratively subdivide this triangle using the loop

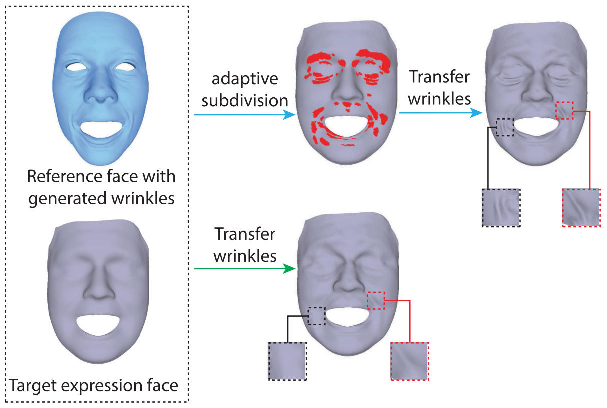


Fig. 7. A comparison of our method with/without wrinkle subdivision. Our method with wrinkle subdivision (blue arrows) can better preserve wrinkle details on the resulting face than our method without wrinkle subdivision (green arrow). The subdivided triangles are visualized in red.

subdivision method [55]. To this end, we can obtain the subdivided target expression face, $T^{'}_{e}$.

Then, we first deform $T^{'}_{e}$ into the above reconstructed face $\hat{S}_e$ to obtain $\hat{T}^{'}_{e}$; and we also deform $T^{'}_{e}$ into $\hat{S}_{e+w}$ to obtain $\hat{T}^{'}_{e+w}$. After that, we solve a similar optimization problem as in Eq. (11). For clarity, we describe it as follows:

$$\arg\min_{\hat{v}_1\ldots\hat{v}_n} \sum_i \|M'_{s_i} - M'_{t_i}\|^2. \tag{14}$$

Different from Eq. (11), in Eq. (14), M'_{s_i} denotes the affine transformation matrices of all the triangles between $\hat{T}^{'}_{e}$ and $\hat{T}^{'}_{e+w}$; and M'_{t_i} denotes the affine transformation matrices of all the triangles between $T^{'}_{e}$ and the to-be-solved target expression face with the wrinkles $T^{'}_{e+w}$ (with the new vertex positions $\hat{v}_1 \ldots \hat{v}_n$). $T^{'}_{e+w}$ is the final output of our approach.

Fig. 7 shows a comparison of our method with/without wrinkle subdivision. As shown in this figure, our method with wrinkle subdivision can better preserve the wrinkle details on the resulting face than our method without wrinkle subdivision. In addition, Fig. 8 shows the intermediate faces generated by our approach during this process.

Wrinkle Transfer for Animation. If we directly apply the above method to transfer wrinkles between two animated faces, temporal inconsistency of the wrinkles between consecutive frames could occur (refer to Fig. 9a). The reason is that the above optimization equation (Eq. (14)) is independently solved for each frame, which cannot guarantee the continuity of the solution across consecutive frames. To address this issue, we change the above optimization equation to the following Eq. (15) that considers not only the wrinkle transfer objective function but also the transferred wrinkle consistency between two consecutive frames (the second regularization term in Eq. (15))

$$\arg\min_{\hat{v}_1\ldots\hat{v}_n} \sum_i \|M'_{s_i} - M'_{t_i}\|^2 + \lambda \sum_i \|M'_{t_{i-1}} - M'_{t_i}\|^2. \tag{15}$$

Here $M'_{t_{i-1}}$ denotes the obtained affine transformation matrices at the previous frame. In our experiments, we experimentally set the weight $\lambda = 0.2$.

Also, in order to ensure the same topology of the subdivided frames in an animation sequence, when making a subdivision decision for a triangle, as long as the maximum of the wrinkle displacements on the three vertices of the same triangle in all the animation frames is larger than a threshold (that is, $\max_{i=1\ldots|F|}(d_{v_{i1}}, d_{v_{i2}}, d_{v_{i3}}) > \epsilon$), we will do subdivision for this triangle in all the animation frames. In this way, the subdivided faces in all the animation frames are guaranteed to have the same topology. Fig. 9 shows a wrinkle transfer comparison for animations using Eq. (15) and using Eq. (14). From this figure, we can observe the result based on Eq. (15) can produce more temporally consistent wrinkles than the result based on Eq. (14).

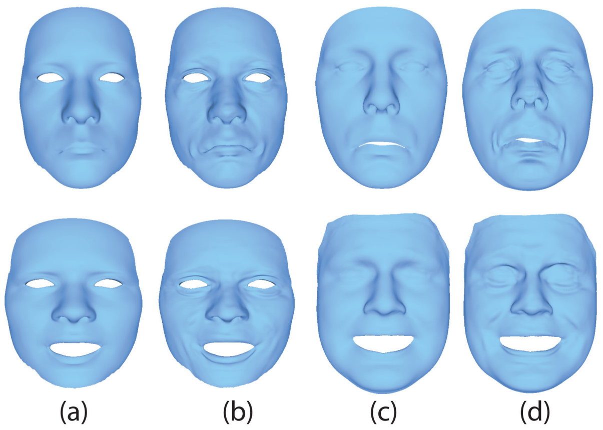


Fig. 8. The intermediate faces generated by our approach during the wrinkle transfer process. Column (a) shows the reference face $\hat{S}_e$; Column (b) shows the reference face with wrinkles, $\hat{S}_{e+w}$; Column (c) shows the target expression face, T_e; and Column (d) shows the final transfer result, $\widetilde{T}^{'}_{e+w}$.

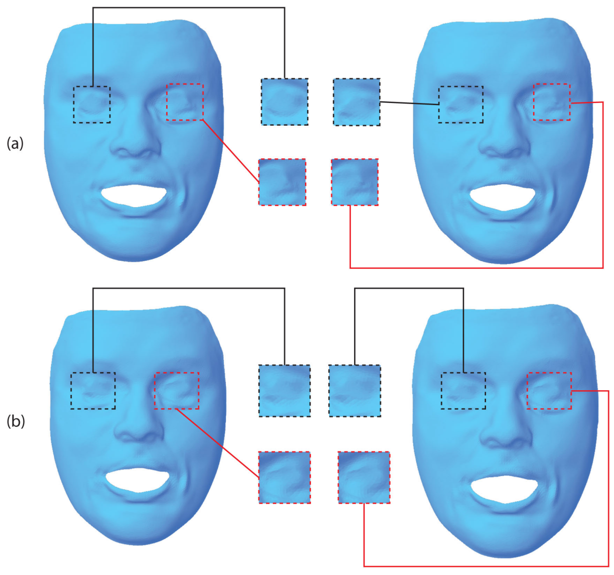


Fig. 9. A wrinkle transfer comparison between using Eq. (14) and using Eq. (15). (a) shows the wrinkles on two consecutive frames are transferred using Eq. (14), (b) shows the wrinkles on the same two continuous frames are transferred using Eq. (15).

7 Results

In this section, we show experimental results by our approach. First, we show novel fine-scale wrinkle generation results on coarse-scale faces with different shapes and expressions. The used coarse-scale faces are directly reconstructed from our bilinear face model. Second, we show wrinkle

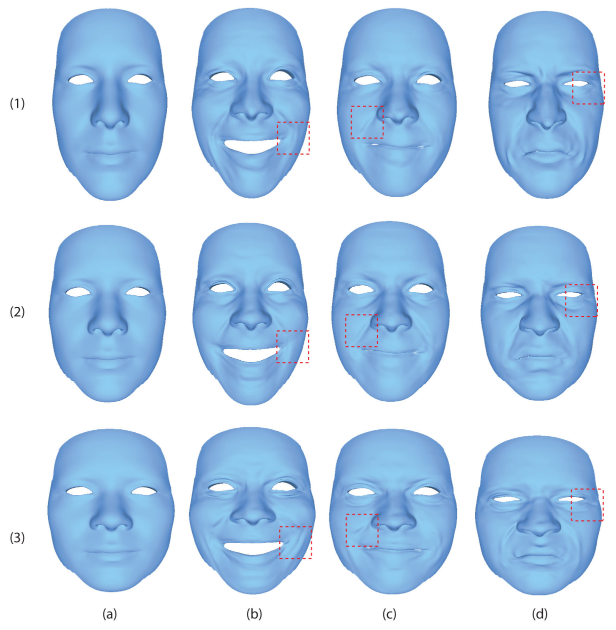


Fig. 10. Wrinkle synthesis results by our approach on target faces with different shapes and expressions. The column (a) shows the target neutral face for each row, and each column from (b)-(d) shows the synthesized wrinkles for different target faces with the same expressions. Each row from (1) to (3) shows the synthesized wrinkles for different expressions on the same target face. As highlighted in the red boxes, for the same expression in the same facial regions, our method can automatically generate distinct wrinkle patterns for different target faces.

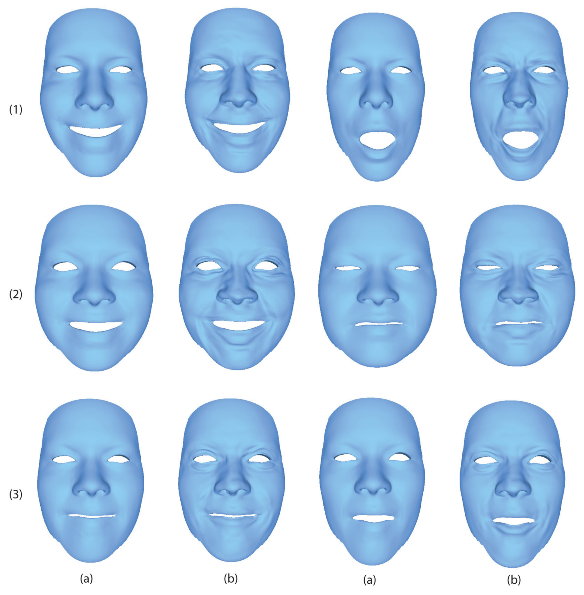


Fig. 11. Comparisons of coarse-scale target expression faces (a) and the corresponding expression faces augmented with the synthesized fine-scale wrinkles by our approach (b). Each row shows a different target face. From each (a)-(b) pair, we can clearly see the target expression faces augmented with our synthesized fine-scale wrinkles are visually more realistic and expressive than the corresponding coarse-scale faces.

transfer results by our approach, that is, transfer synthesized fin-scale wrinkles from the reference face to various coarse-scale target face models that cannot be reconstructed from our bilinear face model.

Fig. 10 shows some generated wrinkle results on coarse-scale faces with different shapes and expressions. Note that all the coarse-scale faces in this figure are reconstructed using our bilinear face model. In this figure, each row shows the synthesized wrinkles for different expressions on the same target face, and each column shows the synthesized wrinkles for different target faces with the same expression. As highlighted in the red boxes, for the same expression in the same facial regions, our method can automatically generate distinct wrinkle patterns for different target faces.

Fig. 11 shows comparisons between the coarse-scale expression faces and the corresponding expression faces augmented with synthesized fine-scale wrinkles by our approach. From each column pair (i.e., (a) versus (b)), we can clearly observe that the expression face augmented with the synthesized fine-scale wrinkles by our approach appear more visually realistic and expressive.

Fig. 12 shows some wrinkle transfer results by our approach. If the given target 3D faces cannot be reconstructed from our FaceWareHouse bilinear face model, we used our wrinkle transfer pipeline to transfer the synthesized wrinkles to such target faces. In this figure, the row (1) shows wrinkle transfer results for different expressions on a given target face model obtained from Paysan *et al.* [54], and the row (2) show wrinkle transfer results for different expressions on a given target face obtained from Huber *et al.* [5]. Both of the target faces in this case cannot be reconstructed from the FaceWareHouse bilinear face model [4]. Each (a)-(b) column pair shows the comparison between a coarse-scale target expression face and the corresponding one augmented with the transferred wrinkles by our approach.

Fig. 13 shows multiple synthesized wrinkle results by our approach given an input target expression face. Specifically, to generate multiple wrinkle results by using our WEN network, we added small random noise to the feature vector of the input target expression face. Therefore, our approach can generate multiple plausible wrinkles for the user to make a selection.

Fig. 14 shows the results of two randomly selected faces that are outliers of the distribution of our WEN model. It shows that our method has limited generalization capability to generate wrinkle details on the outlier faces. In these case, our approach can generate some wrinkles around the eyes and mouth regions.

Wrinkle Interpolation. Since our method can encode wrinkle tensors into a compact latent space, new wrinkle tensors can also be obtained via interpolation. Specifically, given multiple face feature vectors, we first compute the corresponding latent vectors in the latent space. After that, an interpolated vector can be generated via the weighted averaging of the latent vectors. Finally, we can generate the corresponding interpolated wrinkle tensor by using the WEN decoder. Fig. 18 shows bi-linear wrinkle interpolation results. The faces at the positions (1,1) and (1,5), as well as the positions (5,1) and (5,5), have the same identity but different expressions. The faces at the positions (1,1) and (5,1), as well as the positions (1,5) and (5,5), have the same expression but different identities. Rows (2)-(4) and columns (2)-(4) show interpolated results.

7.1 Comparisons

We also compared the results by our approach with the ground truth and those by existing approaches. Fig. 15 shows

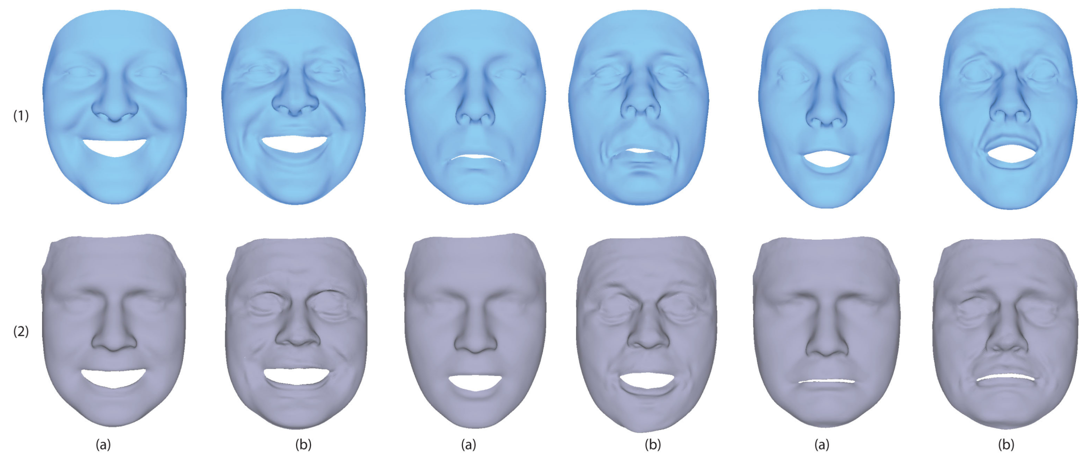


Fig. 12. Wrinkle transfer results by our approach. The target expression faces in the row (1) are obtained from Paysan *et al.* [54], and the target expression faces in the row (2) are obtained from Huber *et al.* [5]. Each (a)-(b) column pair shows the comparison between a coarse-scale target expression face and the corresponding one augmented with the transferred wrinkles by our approach.

the results of three randomly selected faces in the test set. It shows that our method is able to reconstruct almost the same facial wrinkles, compared to the original extracted wrinkles by [10]. Also, it shows that our WEN model encodes the wrinkle tensors into a compact space, and plausible new wrinkle, as shown in this figure, can be well sampled and interpolated.

Fig. 16 shows that our method can even produce results with less artifacts than [10] for some cases. The method in [10] could produce some artifacts when the input video clips have non-frontal head poses, fast head movement, weak lighting, etc. Also, users have to adjust multiple parameters to produce satisfactory results, which is time-consuming and labor-intensive. In contrast, our method can generate realistic wrinkles through sampling and interpolating the latent space in seconds, without the above limitations.

Fig. 17 shows comparison examples between our method and the ground truth data [19]. It shows our method can produce plausible wrinkle details, although our method cannot produce the same wrinkles as the ground truth.

7.2 User Study

We conducted a user study to evaluate the results produced by our proposed wrinkle synthesis and transfer framework. We recruited 20 participants (7 female and 13 male; 23 to 37 years old) to participate in our user study. 6 faces with wrinkles synthesized by our approach (in Fig. 12) were shown to each participant one by one. For each shown face, the participants need to answer the following questions using a 5-likert scale: 1 (not at all), 2 (slightly), 3 (normally), 4 (well enough), and 5 (pretty well).

1) *Expressiveness*: How expressive are the wrinkles on the face?
2) *Matchedness*: How are the wrinkles on the face matched with face shape and expression?
3) *Realism*: How realistic are the wrinkles?
4) *Overall*: How do you evaluate the overall quality of the wrinkles on the face?
5) *Coherence*: How spatially and temporally coherent are the wrinkles on the face?

Fig. 19 shows the user study result. From this figure, we can see most (more than 90 percent) of the participants

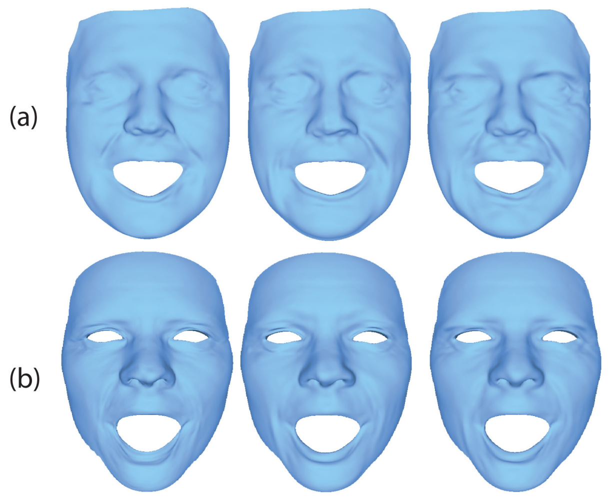


Fig. 13. Three different wrinkle results are generated by our approach based on an input target expression face. The row (a) shows the generated wrinkles on the target expression face, and the row (b) shows the same generated wrinkles on the reference face.

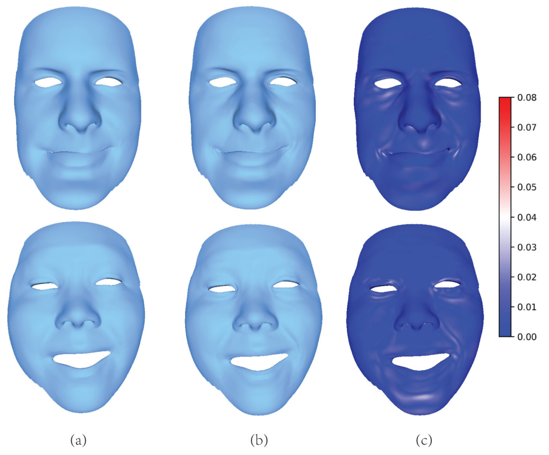


Fig. 14. Example results of our approach when the input faces are outliers of the distribution of our WEN model. Column (a) shows the randomly selected, outlier faces; Column (b) shows the resulting faces by our approach; Column (c) shows the corresponding heat-maps (i.e., (b) column - (a) column) to illustrate the wrinkles.

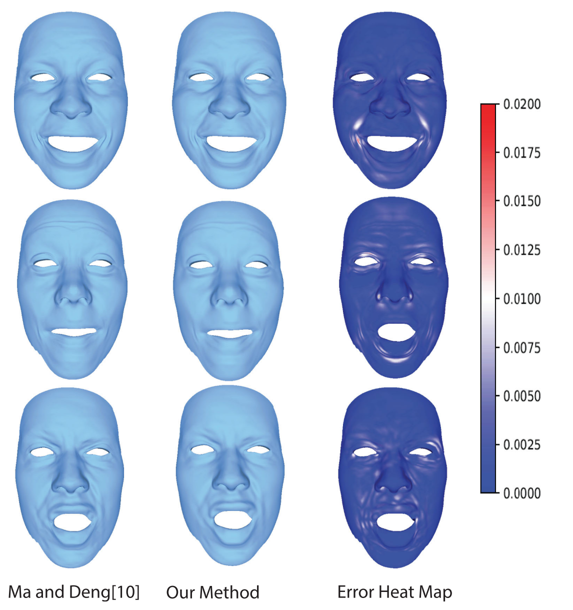


Fig. 15. The result on three randomly selected faces in the test set by our method, compared to the original extracted wrinkles by [10].

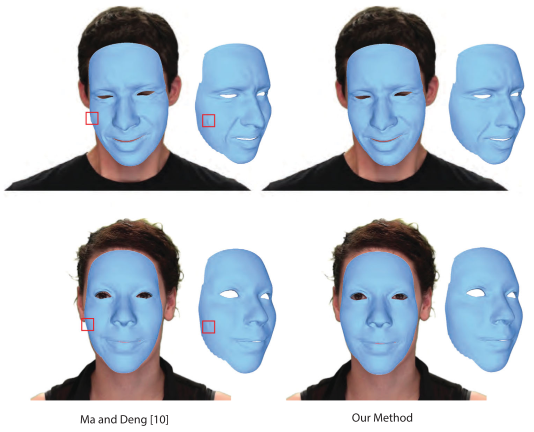


Fig. 16. Examples that show our method can produce wrinkles with less artifacts, compared to the work of [10]. The red rectangle highlights the artifact around the face boundary.

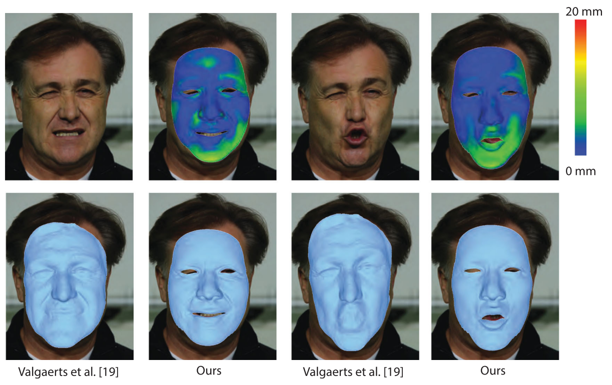


Fig. 17. Comparisons between our method and the reconstructed mesh (considered as the ground-truth) by the binocular method [19]. The top row shows the input image and the corresponding per-vertex error map. The bottom row shows the mesh comparisons.

gave either 4 (well enough) or 5 (pretty well) ratings on the synthesized wrinkles, in terms of all the five measures. We also show the mean and standard deviations of

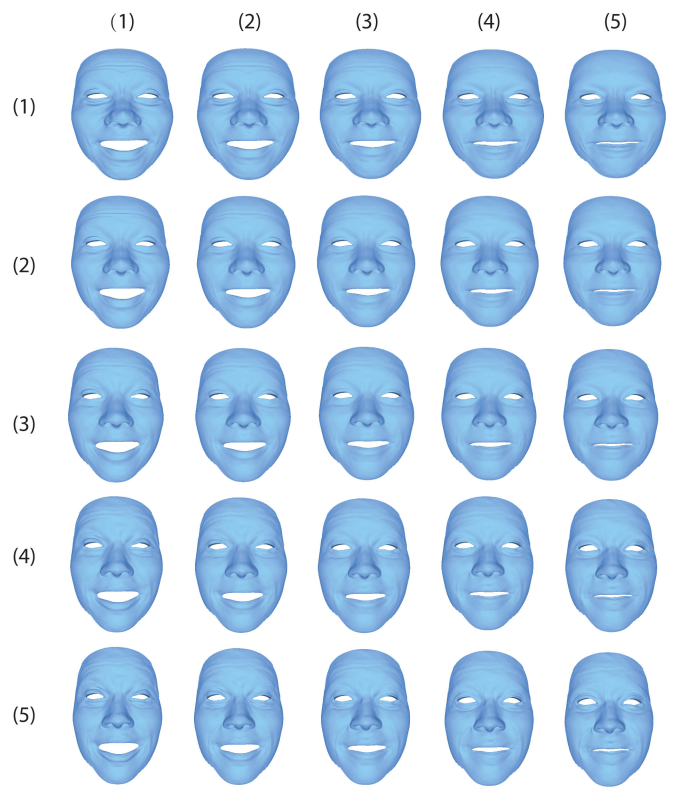


Fig. 18. Bi-linear wrinkle interpolation results. The faces at the positions (1,1) and (1,5), as well as the positions (5,1) and (5,5), have the same identity but different expressions. The faces at the position (1,1) and (5,1), as well as the positions (1,5) and (5,5), have the same expression but different identities.

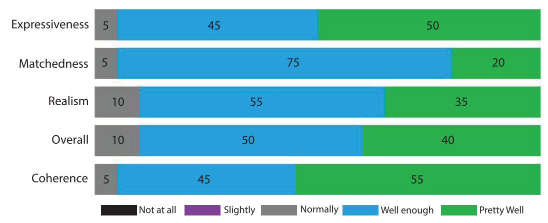


Fig. 19. Obtained votes count for our five questions. The numbers on the bars represent percentile.

TABLE 3
The Mean/Standard Deviation and Median Scores Obtained in the Study

Measure	mean $\pm$ SD	median
Expressiveness	$\mathbf{4.45} \pm 0.59$	**4.5**
Matchedness	4.15 ± 0.48	4.0
Realism	4.25 ± 0.62	4.0
Overall	4.3 ± 0.64	4.0
Coherence	$\mathbf{4.5} \pm 0.59$	**5.0**

the obtained ratings in Table 3. The average scores in terms of all the five measures are more than 4 (well enough). In particular, our methods obtained about 4.5 mean/median scores on both expressiveness and coherence measures. The study outcomes validate that our method can synthesize plausible wrinkles on given coarse-scale faces.

8 Discussion and Conclusion

In this paper we present a VAE-based generative model to automatically synthesize fine-scale geometric wrinkles on novel coarse-scale 3D faces. We also introduce a complete pipeline to transfer fine-scale wrinkles between 3D faces with different topologies and shapes.

Our approach has the following limitations. First, since semantic meaning is not associated to each element of the latent vector in our model. Thus, our approach cannot provide users high-level intuitive controls over the generation of fine-scale wrinkles (e.g., control the shapes and positions of wrinkles). Second, our method synthesizes wrinkles based on the geometric shapes and expressions of target faces. However, in reality, besides the shape and expression, the wrinkles on the human face can also depend on other factors including age and gender. Third, as a data-driven method, the effectiveness of our method is essentially determined by the employed training dataset. Since our used training dataset only includes 50 different identities, if the shape of an input target face is significantly deviated away from the distribution of the face shapes in the training data, our model may generate less satisfactory wrinkles. Also, the training wrinkle data used in this work were extracted from monocular videos. Therefore, the extract wrinkles may not be as accurate as those by multi-view capture methods.

In the future, we plan to add novel high-level user controls over the wrinkle generation process as well as intuitively edit the synthesized or transferred geometric wrinkles on 3D faces. Such systems would be practically useful for numerous applications.

Acknowledgments

This work was supported in part by US NSF IIS-1524782 and NSF IIS-2005430. Huikun Bi was supported in part by the National Natural Science Foundation of China under Grant 62002345.

References

[1] Z. Deng and J. Noh, *Computer Facial Animation: A Survey*. London, U.K.: Springer, 2008, pp. 1–28.
[2] J. P. Lewis, K. Anjyo, T. Rhee, M. Zhang, F. Pighin, and Z. Deng, "Practice and theory of blendshape facial models," in *Proc. Eurographics State Art Rep.*, 2014, pp. 199–218.
[3] M. Zollhöfer *et al.*, "State of the art on monocular 3D face reconstruction, tracking, and applications," *Comput. Graph. Forum*, vol. 37, pp. 523–550, 05 2018.
[4] C. Cao, Y. Weng, S. Zhou, Y. Tong, and K. Zhou, "FaceWarehouse: A 3D facial expression database for visual computing," *IEEE Trans. Vis. Comput. Graphics*, vol. 20, no. 3, pp. 413–425, Mar. 2014.
[5] P. Huber *et al.*"A multiresolution 3D morphable face model and fitting framework," in *Proc. 11th Int. Joint Conf. Comput. Vis. Imag. Comput. Graph. Theory Appl.*, 2016, pp. 79–86.
[6] T. Li, T. Bolkart, M. J. Black, H. Li, and J. Romero, "Learning a model of facial shape and expression from 4D scans," *ACM Trans. Graph.*, vol. 36, no. 6, 2017, Art. no. 194.
[7] P. Garrido, L. Valgaerts, C. Wu, and C. Theobalt, "Reconstructing detailed dynamic face geometry from monocular video," *ACM Trans. Graph.*, vol. 32, no. 6, 2013, Art. no. 158.
[8] P. Garrido *et al.*, "Reconstruction of personalized 3D face rigs from monocular video," *ACM Trans. Graph.*, vol. 35, no. 3, 2016, Art. no. 28.
[9] F. Shi, H.-T. Wu, X. Tong, and J. Chai, "Automatic acquisition of high-fidelity facial performances using monocular videos," *ACM Trans. Graph.*, vol. 33, no. 6, 2014, Art. no. 222.
[10] L. Ma and Z. Deng, "Real-time hierarchical facial performance capture," in *Proc. ACM SIGGRAPH Symp. Interactive 3D Graph. Games*, 2019, pp. 11:1–11:10.
[11] H.-J. Kim, A. C. Öztireli, I.-K. Shin, M. Gross, and S.-M. Choi, "Interactive generation of realistic facial wrinkles from sketchy drawings," *Comput. Graph. Forum*, vol. 34, pp. 179–191, 2015.
[12] L. Zhang, N. Snavely, B. Curless, and S. M. Seitz, "Spacetime faces: High resolution capture for modeling and animation," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 548–558, Aug. 2004.
[13] B. Bickel, M. Botsch, W. R. Angst, H. Pfister, and M. Gross, "Multiscale capture of facial geometry and motion," *ACM Trans. Graph.*, vol. 36, pp. 33–41, Jul. 2007.
[14] H.-D. Huang, J. Chai, X. Tong, and H.-T. Wu, "Leveraging motion capture and 3D scanning for high-fidelity facial performance acquisition," *ACM Trans. Graph.*, vol. 30, Jul. 2011, Art. no. 74.
[15] T. Beeler, B. Bickel, P. Beardsley, B. Sumner, and M. Gross, "High-quality single-shot capture of facial geometry," *ACM Trans. Graph.*, vol. 29, no. 4, Jul. 2010, Art. no. 40.
[16] T. Beeler *et al.*, "High-quality passive facial performance capture using anchor frames," *ACM Trans. Graph.*, vol. 30, no. 4, Jul. 2011, Art. no. 75.
[17] D. Bradley, W. Heidrich, T. Popa, and A. Sheffer, "High resolution passive facial performance capture," in *Proc. ACM SIGGRAPH Papers*, 2010, Art. no. 41.
[18] P. Gotardo, J. Riviere, D. Bradley, A. Ghosh, and T. Beeler, "Practical dynamic facial appearance modeling and acquisition," *ACM Trans. Graph.*, vol. 37, no. 6, Dec. 2018, Art. no. 232.
[19] L. Valgaerts, C. Wu, A. Bruhn, H.-P. Seidel, and C. Theobalt, "Lightweight binocular facial performance capture under uncontrolled lighting," *ACM Trans. Graph.*, vol. 31, no. 6, Nov. 2012, Art. no. 187.
[20] G. Fyffe, A. Jones, O. Alexander, R. Ichikari, and P. Debevec, "Driving high-resolution facial scans with video performance capture," *ACM Trans. Graph.*, vol. 34, no. 1, 2014, Art. no. 8.
[21] S. Suwajanakorn, I. Kemelmacher-Shlizerman, and S. M. Seitz, "Total moving face reconstruction," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 796–812.
[22] S. Suwajanakorn, S. M. Seitz, and I. Kemelmacher-Shlizerman, "What makes tom hanks look like tom hanks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 3952–3960.
[23] T. Weise, H. Li, L. Van Gool, and M. Pauly, "Face/off: Live facial puppetry," in *Proc. ACM SIGGRAPH/Eurographics Symp. Comput. Animation*, 2009, pp. 7–16.
[24] T. Weise, S. Bouaziz, H. Li, and M. Pauly, "Realtime performance-based facial animation," *ACM Trans. Graph.*, vol. 30, no. 4, Jul. 2011, Art. no. 77.
[25] M. Zollhöfer *et al.*, "Real-time non-rigid reconstruction using an RGB-D camera," *ACM Trans. Graph.*, vol. 33, no. 4, Jul. 2014, Art. no. 156.
[26] J. Thies, M. Zollhöfer, M. Nießner, L. Valgaerts, M. Stamminger, and C. Theobalt, "Real-time expression transfer for facial reenactment," *ACM Trans. Graph.*, vol. 34, no. 6, 2015, Art. no. 183.
[27] C. Cao, Q. Hou, and K. Zhou, "Displaced dynamic expression regression for real-time facial tracking and animation," *ACM Trans. Graph.*, vol. 33, no. 4, Jul. 2014, Art. no. 43.
[28] C. Cao, D. Bradley, K. Zhou, and T. Beeler, "Real-time high-fidelity facial performance capture," *ACM Trans. Graph.*, vol. 34, no. 4, Jul. 2015, Art. no. 46.
[29] M. Sela, E. Richardson, and R. Kimmel, "Unrestricted facial geometry reconstruction using image-to-image translation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 1585–1594.
[30] Y. Guo, J. Zhang, J. Cai, B. Jiang, and J. Zheng, "CNN-based real-time dense face reconstruction with inverse-rendered photo-realistic face images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 6, pp. 1294–1307, Jun. 2019.
[31] C. R. Qi, H. Su, M. Nießner, A. Dai, M. Yan, and L. J. Guibas, "Volumetric and multi-view CNNs for object classification on 3D data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 5648–5656.
[32] M. Tatarchenko, A. Dosovitskiy, and T. Brox, "Multi-view 3D models from single images with a convolutional network," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 322–337.
[33] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2013, *arXiv:1312.6114*.
[34] A. Brock, L. Theodore, J. M. Ritchie, and N. Weston, "Generative and discriminative voxel modeling with convolutional neural networks," 2016, *arXiv:1608.04236*.

[35] C. Nash and C. K. I. Williams, "The shape variational autoencoder: A deep generative model of part-segmented 3D objects," *Comput. Graph. Forum*, vol. 36, no. 5, pp. 1–12, Aug. 2017.

[36] J. Li, K. Xu, S. Chaudhuri, E. Yumer, H. Zhang, and L. Guibas, "GRASS: Generative recursive autoencoders for shape structures," *ACM Trans. Graph.*, vol. 36, pp. 1–14, Jul. 2017.

[37] Q. Tan, L. Gao, Y.-K. Lai, and S. Xia, "Variational autoencoders for deforming 3D mesh models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5841–5850.

[38] A. Ranjan, T. Bolkart, S. Sanyal, and M. J. Black, "Generating 3D faces using convolutional mesh autoencoders," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 704–720.

[39] S. Saito, L. Hu, C. Ma, H. Ibayashi, L. Luo, and H. Li, "3D hair synthesis using volumetric variational autoencoders," *ACM Trans. Graph.*, vol. 37, no. 6, Dec. 2018, Art. no. 208.

[40] L. Gao *et al.*, "SDM-NET: Deep generative network for structured deformable mesh," *ACM Trans. Graph.*, vol. 38, no. 6, Nov. 2019, Art. no. 243.

[41] K. Mo *et al.*, "StructureNet: Hierarchical graph networks for 3D shape generation," 2019, *arXiv: 1908.00575*.

[42] Y. Zhang, E. C. Prakash, and E. Sung, "Real-time physically-based facial expression animation using mass-spring system," in *Proc. Comput. Graph. Int.*, 2001, pp. 347–350.

[43] Y. Zhang, E. C. Prakash, and E. Sung, "A new physical model with multilayer architecture for facial expression animation using dynamic adaptive mesh," *IEEE Trans. Vis. Comput. Graphics*, vol. 10, no. 3, pp. 339–352, May/Jun. 2004.

[44] Y. Zhang, T. Sim, C. L. Tan, and E. Sung, "Anatomy-based face reconstruction for animation using multi-layer deformation," *J. Vis. Lang. Comput.*, vol. 17, no. 2, pp. 126–160, 2006.

[45] P. Li and P. G. Kry, "Multi-layer skin simulation with adaptive constraints," in *Proc. 7th Int. Conf. Motion Games*, 2014, pp. 171–176.

[46] L. Boissieux, G. Kiss, N. M. Thalmann, and P. Kalra, "Simulation of skin aging and wrinkles with cosmetics insight," in *Proc. Comput. Animation Simul.*, 2000, pp. 15–27.

[47] K. Venkataraman, S. Lodha, and R. Raghavan, "Technical section: A kinematic-variational model for animating skin with wrinkles," *Comput. Graph.*, vol. 29, no. 5, pp. 756–770, Oct. 2005.

[48] Y. Wang, C. C. Wang, and M. M. Yuen, "Fast energy-based surface wrinkle modeling," *Comput. Graph.*, vol. 30, no. 1, pp. 111–125, 2006.

[49] L. D. Cutler *et al.*, "An art-directed wrinkle system for CG character clothing and skin," *Graphical Models*, vol. 69, no. 5/6, pp. 219–230, 2007.

[50] W.-C. Ma *et al.*, "Facial performance synthesis using deformation-driven polynomial displacement maps," in *Proc. ACM SIGGRAPH Asia Papers*, 2008, Art. no. 121.

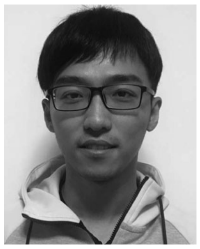
[51] R. W. Sumner and J. Popović, "Deformation transfer for triangle meshes," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 399–405, Aug. 2004.

[52] S. R. Livingstone and F. A. Russo, "The Ryerson audio-visual database of emotional speech and song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English," *PloS One*, vol. 13, no. 5, 2018, Art. no. e0196391.

[53] M. Pantic, M. Valstar, R. Rademaker, and L. Maat, "Web-based database for facial expression analysis," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2005, pp. 5–pp.

[54] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter, "A 3D face model for pose and illumination invariant face recognition," in *Proc. 6th IEEE Int. Conf. Adv. Video Signal Based Surveillance*, 2009, pp. 296–301.

[55] C. Loop, "Smooth subdivision surfaces based on triangles," Master's thesis, Dept. Math., Univ. Utah, Salt Lake City, UT, USA, 1987.



Qixin Deng received the BS degree in electronics and information engineering from Zhengzhou University, Zhengzhou, China, in 2014, and the MSc degree in electrical and computer engineering from Purdue University, West Lafayette, Indiana, in 2017. He is currently working toward the PhD degree with the Department of Computer Science, University of Houston, Houston, Texas. His research interests include face modeling, animation, and machine learning.

Luming Ma received the MS degree in computer science from Bridgeport University, Bridgeport, Connecticut, in 2008. He is currently working toward the PhD degree with the University of Houston, Houston, Texas. His research interests include real-time facial reconstruction and manipulation from RGB video.

Aobo Jin received the BS degree in electrical engineering from the Dalian University of Technology, Dalian, China, in 2011, and the MSc degree in electrical engineering from the Department of Electrical and Computer Engineering, University of Houston, Houston, Texas, in 2016. He is currently working toward the PhD degree with the Department of Computer Science, University of Houston, Houston, Texas. His research interests include computer graphics, virtual human modeling and animation, and sketch-based modeling.

Huikun Bi received the PhD degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China and the University of Chinese Academy of Sciences, Beijing, China. She is currently an assistant professor with the Institute of Computing Technology, Chinese Academy of Sciences. Her research interests include crowd simulation and human behavior prediction.

Binh Huy Le received the BS degree in computer science from the Vietnam National University, Hanoi, Vietnam, in 2008 and the PhD degree in computer science from the Department of Computer Science, University of Houston, Houston, Texas, in 2014. He is currently a research scientist with Search for Extraordinary Experiences Division, Electronic Arts Inc. His research interests include computer graphics, computer animation, and virtual humans.

Zhigang Deng (Senior Member, IEEE) received the BS degree in mathematics from Xiamen University, Xiamen, China, the MS degree in computer science from Peking University, Beijing, China, and the PhD degree in computer science from the Department of Computer Science, University of Southern California, Los Angeles, California, in 2006. He is currently a full professor of computer science with the University of Houston. His research interests include computer graphics, computer animation, and virtual human modeling and animation.