

Dynamic Facial Asset and Rig Generation from a Single Scan

JIAMAN LI*, University of Southern California and USC Institute for Creative Technologies

ZHENGFEI KUANG*, University of Southern California and USC Institute for Creative Technologies

YAJIE ZHAO†, USC Institute for Creative Technologies

MINGMING HE, USC Institute for Creative Technologies

KARL BLADIN, USC Institute for Creative Technologies

HAO LI, University of Southern California, USC Institute for Creative Technologies, and Pinscreen

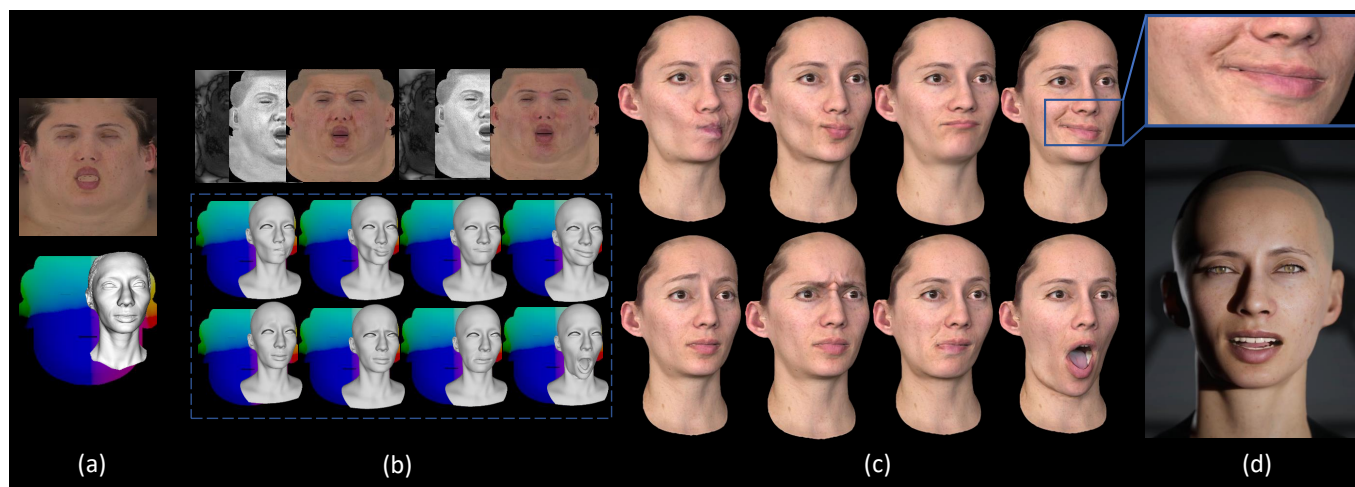


Fig. 1. Given a single neutral scan (a), we generate a complete set of dynamic face model assets, including personalized blendshapes and physically-based dynamic facial skin textures of the input subjects (b). The results carry high-fidelity details which we render in Arnold [Maya 2019] (c). Our generated facial assets are animation-ready as shown in (d).

The creation of high-fidelity computer-generated (CG) characters for films and games is tied with intensive manual labor, which involves the creation of comprehensive facial assets that are often captured using complex hardware. To simplify and accelerate this digitization process, we propose a framework for the automatic generation of high-quality dynamic facial models, including rigs which can be readily deployed for artists to polish. Our framework takes a single scan as input to generate a set of personalized blendshapes, dynamic textures, as well as secondary facial components (e.g., teeth and eyeballs). Based on a facial database with over 4,000 scans

with pore-level details, varying expressions and identities, we adopt a self-supervised neural network to learn personalized blendshapes from a set of template expressions. We also model the joint distribution between identities and expressions, enabling the inference of a full set of personalized blendshapes with dynamic appearances from a single neutral input scan. Our generated personalized face rig assets are seamlessly compatible with professional production pipelines for facial animation and rendering. We demonstrate a highly robust and effective framework on a wide range of subjects, and showcase high-fidelity facial animations with automatically generated personalized dynamic textures.

* indicates equal contribution.

† indicates corresponding author.

Authors' addresses: Jiaman Li, University of Southern California, USC Institute for Creative Technologies; Zhengfei Kuang, University of Southern California, USC Institute for Creative Technologies; Yajie Zhao, USC Institute for Creative Technologies; Mingming He, USC Institute for Creative Technologies; Karl Bladin, USC Institute for Creative Technologies; Hao Li, University of Southern California, USC Institute for Creative Technologies, Pinscreen.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2020 Copyright held by the owner/author(s). Publication rights licensed to ACM.

0730-0301/2020/12-ART215 \$15.00

<https://doi.org/10.1145/3414685.3417817>

CCS Concepts: • **Computer methodologies** → **Face Animation**.

Additional Key Words and Phrases: Face Rigging, Blendshapes, Animation, Physically-Based Face Rendering, Performance Capture, Deformation Transfer.

ACM Reference Format:

Jiaman Li, Zhengfei Kuang, Yajie Zhao, Mingming He, Karl Bladin, and Hao Li. 2020. Dynamic Facial Asset and Rig Generation from a Single Scan. *ACM Trans. Graph.* 39, 6, Article 215 (December 2020), 17 pages. <https://doi.org/10.1145/3414685.3417817>

1 INTRODUCTION

High-quality and personalized digital humans are relevant to a wide range of applications, such as film and game production (e.g. Unreal Engine, Digital Doug), and virtual reality [Fyffe et al. 2014; Lombardi

et al. 2018; Wei et al. 2019]. To produce high-fidelity digital doubles, complex capture equipment is often needed in conventional computer graphics pipelines, and the acquired data typically undergoes intensive manual post-processing by a production team. New approaches based on deep learning-based synthesis are promising as they show how photorealistic faces can be generated from captured data directly [Lombardi et al. 2018; Wei et al. 2019] allowing one to overcome the notorious Uncanny Valley. In addition to their intensive GPU compute requirements and the need for large volumes of training data, these deep learning-based methods are still difficult to integrate seamlessly into virtual CG environments as they lack relighting capabilities and fine rendering controls, which prevents them from being adopted for games and film production. On the other hand, realistic digital doubles in conventional graphics pipelines require months of production and involve large teams of highly skilled digital artists as well as sophisticated scanning techniques [Ghosh et al. 2011]. Building facial assets of a virtual character typically requires a number of facial expression models often based on the Facial Action Coding System (FACS), as well as physically-based texture assets (e.g., albedo, specular maps, displacement maps) to ensure realistic facial skin reflectance in a virtual environment.

Several recent works have shown how to automate and reduce the effort for generating personalized facial rigs. The works of Laine et al. [2017]; Li et al. [2010]; Ma et al. [2016]; Pawaskar et al. [2013] propose to automatically build personalized blendshapes using a varying number of personalized facial scans. While effective for production pipelines, these methods either require a large number of facial scans as input and considerable post-processing, or they only focus on generating a personalized geometry for the expressions, without the textures. For consumer-accessible avatar creation techniques, the works of Casas et al. [2016]; Hu et al. [2017]; Ichim et al. [2015]; Nagano et al. [2018]; Thies et al. [2016] demonstrate digitization capabilities from video sequences or even a single input image. However, due to the limited input data, the resulting models often lack details or the generated assets do not contain physically-based properties for dynamic expressions. We propose an approach based on a 3D scan as input and our goal is to produce a fully rigged model with fixed topology, personalized blendshapes expressions along with corresponding dynamic and physically-based texture maps. We observe that a large amount of labeled data can enable the learning of personalized models and dynamic deformations such that wrinkle formations are specific to the shape and appearance of the subject. In particular, we extend recent deep learning approaches for high-resolution physically-based skin assets [Li et al. 2020; Yamaguchi et al. 2018], to generate dynamic high-resolution facial texture attributes (albedo, specular maps, and displacement maps), in order to produce effects such as plausible personalized wrinkles during animation. Existing methods transfer facial expression details from a generic database, which may lead to reasonable output for the geometry, but certainly lack dynamic texture variations.

We present a framework to automate and simplify the generation of high-quality facial rig assets, consisting of personalized blendshapes, dynamic physically-based skin attributes (albedo, specular reflection, displacement maps), including secondary facial components (e.g. eyes, teeth, gums, and tongue), from a single neutral

geometric model and albedo map as input. Our generated assets can be directly fed into professional production pipelines. We use a high-fidelity facial scan database [Li et al. 2020] and address both the problems of generating personalized blendshapes and inferring dynamic physically-based skin properties. We first propose an end-to-end self-supervised learning framework to overcome the lack of ground truth data for personalized blendshapes and dynamic textures. By modeling the correlation between identities and personalized expressions on the database with 178 identities, each having 19~26 different captured expressions, we eliminate the requirement of user-specific scans for personalized blendshapes generation using a trade-off between semantic meaning and personality. Our approach uses an intermediate conversion of neutral geometry and 2D textures to a common parameterization in UV space, which enables training and inference of dynamic geometry and texture deformation in a compact form inspired by Li et al. [2020].

Learning is performed using a high-fidelity facial scan dataset with over 4000 scans with pore-level details and different expressions. Our approach can automatically produce personalized blendshapes that reflect personalized expressions of a person from only one neutral scan. We demonstrate the effectiveness of our framework on a wide range of subjects and showcase a number of compelling facial animations.

In summary, our major contributions are as follows:

- We propose an end-to-end framework to automate the generation of high-quality facial assets and rigs. Given a single neutral face scan with albedo as input, we produce plausible personalized blendshapes, secondary facial components (e.g. teeth, eyelashes), and most importantly, physically-based textures that are both dynamic and personalized to the appearance of the input subject.
- We present a novel self-supervised deep learning approach to improve the personalized results using a generic facial expression template model. In particular, our approach can model the joint distribution between individual identities and their expressions in a large high-fidelity face database.
- We also introduce a novel physically-based texture synthesis framework conditioned on neutral geometry and textures. Using a new compress and stretch map approach, we are able to synthesize dynamic expression-specific textures, including albedo, specular, and fine-scale displacement maps.
- We will make our code, models and database with all texture assets public to facilitate further research on automating high-quality avatar generation.

2 RELATED WORK

Facial Capture. Due to increased demands for realistic digital avatars, facial capture and performance capture have been well-studied. Based on a multi-view stereo system, fine-scale details of the captured face can be recovered in a controlled environment with multiple calibrated DSLR cameras as in the work of Beeler et al. [2010]. A more intricate system by Ghosh et al. [2011] extends the view-dependent method [Ma et al. 2007] by adopting fixed linear polarized spherical gradient illumination in front of the cameras

and enables accurate acquisition of diffuse albedo, specular intensity, and pore-level normal maps. Fyffe et al. [2016] later propose a method that employs commodity hardware, while recording comparable results with off-the-shelf components and near-instant capture. Meanwhile, works on passive facial performance capturing [Beeler et al. 2011; Bradley et al. 2010; Fyffe et al. 2014; Valgaerts et al. 2012] have shown impressive detailed results for highly articulated motion. Recently, Gotardo et al. [2018] propose a method to acquire dynamic properties of facial skin appearance, including dynamic diffuse albedo, specular intensity, and normal maps. These methods provide decent training data and set a high baseline for lightweight facial capture and modeling approaches.

Facial Rigging. Creating facial animation is a well-studied problem with a plethora of methods proposed in film and video game industries. Blanz and Vetter [1999] first introduce the Morphable Face Model to represent face shapes and textures of different identities using principal component analysis (PCA) learned from 200 laser scan subjects. Later, the improved parametric face models are built using 10,000 high-quality 3D face scans [Booth et al. 2017, 2016]. A linear model generated from web images has also been demonstrated [Kemelmacher-Shlizerman 2013].

Modeling of variational face expressions using blendshapes is a popular approach in many applications [Thies et al. 2015, 2016]. The approach models facial expressions as activation of shape units represented by a linear basis of facial expression vectors [Lewis et al. 2014]. Amberg et al. [2008] combines a PCA model of a neutral face with a PCA space derived from the residual vectors of different expressions to the neutral pose. Blendshapes can either be hand-crafted by animators [Alexander et al. 2009; Olszewski et al. 2016], or be generated via statistical analysis from large facial expression datasets [Cao et al. 2014; Li et al. 2017; Vlastic et al. 2005]. The multi-linear model [Cao et al. 2014; Vlastic et al. 2005] offers a way of capturing a joint space of expression and identity. Li et al. [2017] propose the FLAME model learned from thousands of scans and significantly improve the model expressiveness.

Personalized Blendshape Generation. As an effort to advance and scale the production of facial animations, expression cloning [Noh and Neumann 2001] has been introduced to mimic the existing deformation of a source 3D face model onto a target face. Sumner and Popović [2004] propose deformation transfer for generic 3D triangle mesh. Onizuka et al. [2019] propose a landmark-guided deformation transfer method to generate expressions for any target avatar that directly maps to a generic blendshape template. These methods can generate an expression for a novel subject but might fail to capture personalized behavior due to the lack of personal information.

To build robust face rigs, we need to reconstruct a dynamic expression model that faithfully captures the subject's specific facial movements. A full set of personalized blendshapes for a specific subject can be built from 3D scan data of the same subject [Carri-gan et al. 2020; Huang et al. 2011; Li et al. 2010; Weise et al. 2009; Zhang et al. 2004]. These methods can reconstruct expressions that capture the target's personal expressions, but a large set of action units or sparse expressions are required as input. Some follow-up works [Bouaziz et al. 2013; Hsieh et al. 2015; Li et al. 2013] apply

expression transfer on top of a generic face model and train model correctives for the expressions during tracking with samples obtained from RGB-D video input. Ichim et al. [2015] and Cao et al. [2016] propose a comprehensive pipeline to generate a dynamic 3D avatars based on personalized blendshapes with a monocular video of a specific expression sequence. Casas et al. [2016] reconstruct blendshapes and each blendshape's textures with a Kinect. Garrido et al. [2016] introduce a video-based method, which makes blendshape generation suitable for legacy video footage.

Deep Face Models. As deep learning-based methods for 3D shapes analysis have attracted increasing attention in recent years, some methods for non-linear 3D Morphable Model learning have been introduced [Bagautdinov et al. 2018; Li et al. 2020; Tewari et al. 2017; Tran et al. 2019; Tran and Liu 2018]. These models are formulated as decoders using convolutional neural networks, some of these methods use fully connected layers or 2D convolutions in the image space [Li et al. 2020], while some build decoders in the mesh domain to exploit the local geometry of 3D structures [Abrevaya et al. 2019; Cheng et al. 2019; Litany et al. 2018; Ranjan et al. 2018; Zhou et al. 2019].

Image-to-Image Translation. Isola et al. [2017] present Pix2Pix, a method to translate images from one domain to another. It consists of a generator and a discriminator, where the objective of the generator is to translate images from domain A to B, while the discriminator aims to distinguish real images from the translated ones. Wang et al. [2018b] later extend this work to Pix2PixHD to synthesize high-resolution photo-realistic images from semantic label maps. Some works [Lee et al. 2019; Wang et al. 2019, 2018a] on the learning of "translation" functions for videos also incorporate a spatio-temporal adversarial objective. Image-to-image translation has also been adopted to generate 3D faces or detailed face textures. Matan Sela [2017] propose a Pix2Vertex framework using image-to-image translation that jointly maps the input image to a depth image and a facial correspondence map. Huynh et al. [2018] applies this image-to-image translation framework to infer mesoscopic facial geometry with high-quality training data captured using the Light Stage. Yamaguchi et al. [2018] presents a comprehensive method to infer facial reflectance maps from unconstrained image input. Nagano et al. [2018] introduces a framework to synthesize arbitrary expressions in image space and textures in UV space from a single input image. Chen et al. [2019] adopts a conditional GAN to synthesize geometric details (*wrinkles*) by estimating a displacement map over a proxy mesh. Similarly, Yang et al. [2020] infers a displacement map on a base mesh generated from a single image based on a large high-quality face dataset.

3 SYSTEM OVERVIEW

Our system takes a single scanned neutral geometry with an albedo map as input and generates a set of face rig assets and texture attributes for physically based production-level rendering. As shown in Fig. 2, we developed a cascaded framework, in which we first estimate a set of personalized blendshape geometries of the input subject using a Blendshape Generation network, followed by a Texture Generation network to infer a set of dynamic maps including

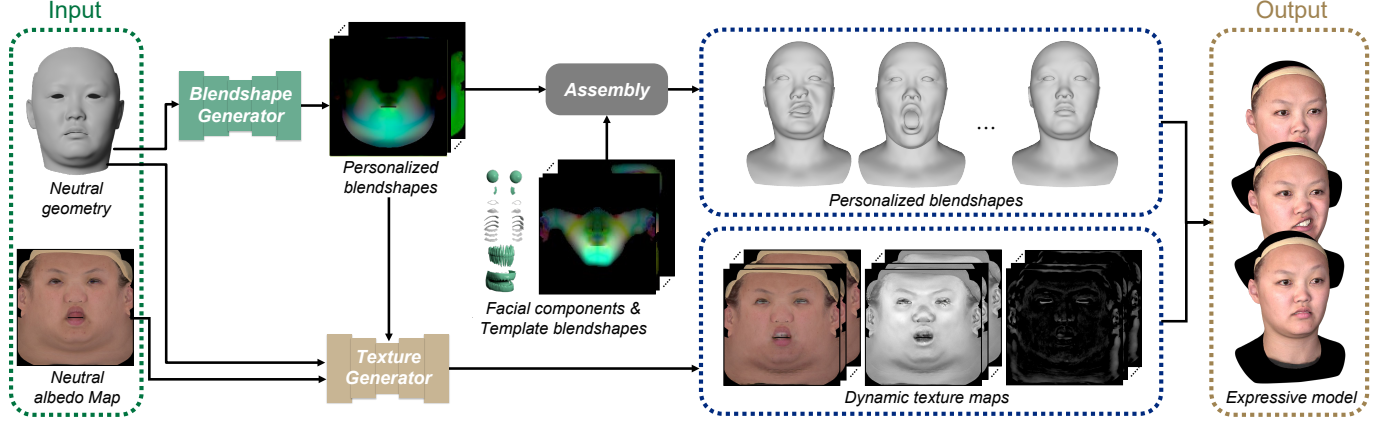


Fig. 2. System Overview. Given the model from a single scan in a neutral expression, the blendshape generation module first generates its personalized blendshapes. Then, using the personalized blendshapes, along with the input neutral model and its albedo map, the texture generation module produces high-resolution dynamic texture maps including albedo, specular intensity and displacement maps. With these assets ready, we then assemble personalized blendshapes and the input neutral model into 3D models, combining other facial components (eyes, teeth, gums, and tongue) from the template models. The final output is complete face models rendered using the blendshape models and textures.

albedo maps, specular intensity maps, and displacement maps. In the final step, we combine the obtained secondary facial components (*i.e.* teeth, gums, and eye assets) from a set of template shapes, to assemble the final face model.

4 BLENDSHAPE GENERATION

Our goal is to automatically generate a full set of personalized blendshapes from a neutral 3D face of a novel subject. This is a challenging problem since generating subject-specific blendshapes usually requires different expressions of the subject. Thanks to our large-scale dataset which consists of various expressions as described in Sec. 7, we introduce a self-supervised pipeline that learns to generate personalized blendshapes based on expressions. Our first task is to imitate the process followed by artists isolating scanned expressions to unit blendshapes using deep neural networks. Given a set of pre-defined generic template blendshapes as a semantic reference and multiple well-defined scan expressions of the same subject, our first goal is to automatically generate the personalized blendshapes of the input subject.

The generic template blendshape model is defined as a generic model S_0 in neutral expression and a set of N (in our case $N = 55$) additive vector displacements $\mathbf{S} = \{S_1, \dots, S_N\}$. Expressions can be generated as $P_k = S_0 + \sum_{i=1}^N \alpha_{ik} S_i$, where α_{ik} are the blending weights for the expression k . For a new subject j , given his/her neutral expression model S_0^j and other expressions P_k^j , their personalized blendshapes S_i^j can be optimized by minimizing the reconstruction loss of P_k^j and ground truth expression P_k^j if blending weights α_{ik}^j , $i = 1, \dots, N$ for P_k^j are known:

$$P_k^{j'} = S_0^j + \sum_{i=1}^N \alpha_{ik}^j S_i^j. \quad (1)$$

This is the foundation of our self-supervised learning scheme.

Based on our template blendshape set, we also pre-defined $k = 26$ FACS expressions for building the dataset (excluding neutral expression). The FACS expressions refer to a set of standardized facial poses that can be performed by a person and generally correspond to a combination of blendshapes (blending weights will be either 0 or 1) with minimum motion overlap and maximum blendshape coverage. We assume that our captured FACS covers all the blendshapes and they can be isolated to unit blendshapes losslessly (more details in Sec. 7). So far, for each of the training subjects, we have a set of captured FACS expressions with corresponding combinations (0 or 1 blending weights). However, it would be irresponsible to say that the blending weights of FACS can be regarded as ground truth for real scans. One can easily perform unwanted motions when trying to express a predefined FACS expression (*e.g.* FACS *smile* consists only *Left_Lip_Corner_Puller* and *Right_Lip_Corner_Puller*, ended with unexpected eye motion captured). To address this issue, we propose a two-stage learning framework as shown in Fig. 3. The Estimation Stage, as the first one, fixes the initial blending weights to generate a set of blendshapes that optimally preserves identity and semantics, while its counterpart, the Tuning Stage, finetunes the initial blendshapes by jointly learning blending weights to better fit captured FACS expressions.

4.1 Estimation Stage

As shown in Fig. 3, the Estimation Stage takes a model with neutral expression S_0^j and pre-defined blending weights for FACS expression P_k^j as its input. It contains of a *Blendshape Generator*, which learns to generate personalized blendshapes that are used to reconstruct the expression P_k^j using Eq. 1. We define a reconstruction loss in Eq. 2 between the reconstructed expression and the input expression.

$$L_{rec} = \sum_{x \in P_k^j} \|P_k^{j'}(x) - P_k^j(x)\|_1. \quad (2)$$

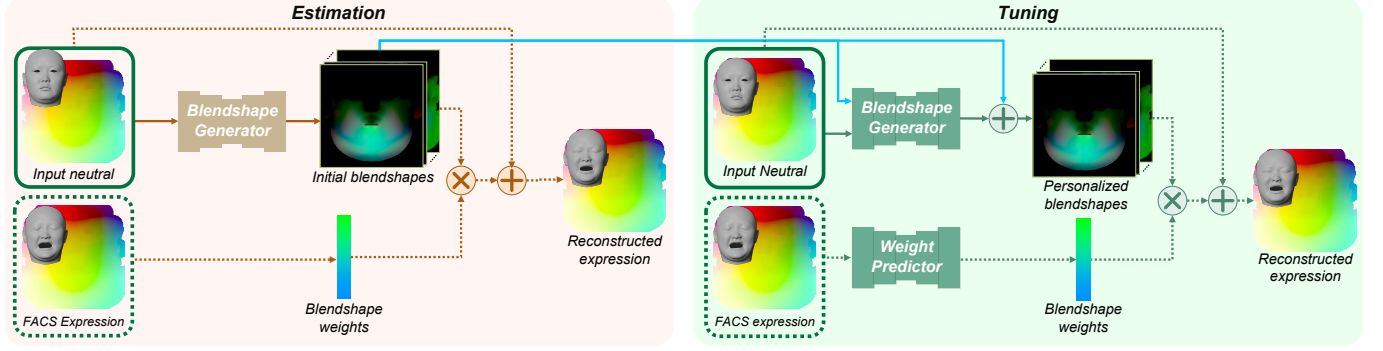


Fig. 3. Two-stage self-supervised learning framework. Given a model in a neutral expression, the Estimation Stage first predicts the initial blendshapes which will work as input for the Tuning Stage to generate the final personalized blendshapes. The inference pipeline is connected by solid lines. The training architecture also involves the parts in dashed lines for computing reconstruction loss. In the Estimation Stage, the *Blendshape Generator* learns to generate the initial blendshapes from the input neutral expression, which combines with the known blending weights to reconstruct the non-neutral expressions. In the Tuning Stage, the *Blending Weight Predictor* is added to predict blending weights for the personalized blendshapes which will be used to reconstruct the input expression.

Inspired by the idea in Li et al. [2010] which emphasizes the importance of relative change between the template and the target models, we propose to learn blendshape offsets instead of blendshapes themselves because: (1) blendshape offsets are distributed in a nearly standard normal distribution which is easy for the network to learn; (2) blendshape offsets can better demonstrate the identity difference. For the example in Fig. 4, the same expression of two different subjects are presented, where their difference is most obviously shown by the blendshape offsets. Thus, the output of the *Blendshape Generator*, $\{\Delta S_1^j, \dots, \Delta S_n^j\}$, are the offsets from the template blendshape to the target, which can be used to reconstruct the target personalized blendshapes by adding the template blendshapes as:

$$S_i^j = \Delta S_i^j + S_i, \forall i \geq 1. \quad (3)$$

To make the target blendshapes semantically consistent with the template blendshapes, we define a regularization term on blendshape offsets to minimize their relative difference.

$$L_{reg} = \sum_{i=1}^N \sum_{x \in S_i} g_i m_i(x) \|\Delta S_i^j(x)\|_1, \forall i \geq 1. \quad (4)$$

where g_i are global weights for different kinds of blendshapes and $m_i(x)$ are local weights for each vertex x in the blendshape S_i , defined as Eq. 5 and Eq. 6.

The global weights are defined as:

$$g_i = \frac{\lambda_g}{\sum_{x \in S_i} \|S_i(x)\|_2}, \forall i \geq 1. \quad (5)$$

where λ_g is a scale factor restricting the maximum g_i to 1. Considering the scale difference in different blendshapes, we introduce global weights to balance the influence of each blendshape for regularization loss. For example, the shape *Jaw_Open* involves more moving vertices than *Left_Eye_Open*. If the same weight is assigned to both, the regularization loss will be dominated by *Jaw_Open*, thus underestimating less pronounced shapes. Thus, we adopt a strategy that assigns a smaller regularization weight to blendshapes with larger offset scale. A similar strategy is used in Chen et al. [2018], where

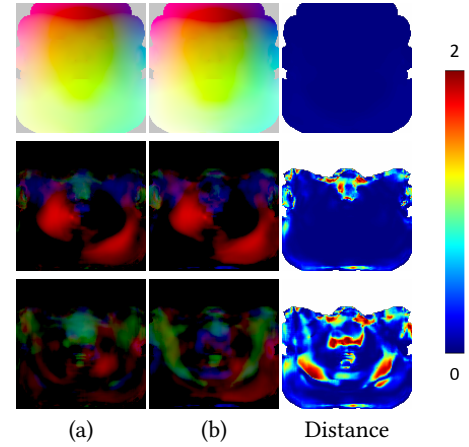


Fig. 4. Visualization of cosine distance maps between expressions, blendshapes and blendshape offsets. (a) and (b) show the same expression of different subjects represented by absolute positions in expression geometry P_i^j (Row 1), blendshape offsets from neutral expression S_i^j (Row 2) and offsets from the template blendshape ΔS_i^j (Row 3). Note that the distance map in Row 1 is almost filled with zeroes. This is because the average difference of the same expression between different individuals is much less than the scale of the human head.

adaptive weights for multi-objective loss are applied to balance the gradients in the training.

The local weights m_i are defined by normalized norms of template blendshapes in which the vertex values are normalized to $(0, 1]$:

$$m_i(x) = \frac{\lambda_l^i}{\|S_i(x)\|_2}, \forall x \in S_i. \quad (6)$$

where λ_l^i is a scale factor restricting the maximum m_i to 1 (excluding fixed vertices), as for fixed vertices in blendshape S_i (where $S_i(x) = 0$), we manually assign a relative large weight to constrain

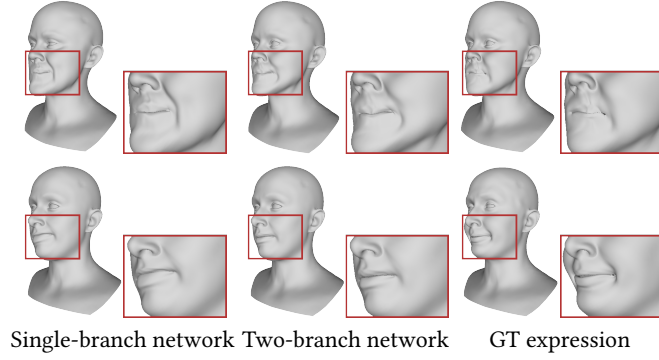


Fig. 5. Comparison of two blendshape models generated by the *Blendshape Generator* with a single-branch network and a two-branch network in the Estimation Stage. GT expression represents the reference FACS expression which is most semantically similar to the corresponding blendshape. Compared to the single-branch results, the two-branch results are more similar to the reference FACS expressions while keeping the semantic meaning of the generic blendshapes.

their movements (we used 4 in our experiments). For each blendshape, the changes from the input neutral face are dominated by only a subset of vertices while the remaining vertices remain unchanged. The local weights are used to penalize large movements of the unchanged vertices and ensure the overall isolation of the generated blendshapes.

Finally, we combine the reconstruction loss L_{rec} and the regularization term L_{reg} to yield the loss function for the *Blendshape Generator*:

$$L_G = L_{rec} + \omega_{reg} L_{reg}, \quad (7)$$

where ω_{reg} is the regularization weight which is set to 1 in the training.

The *Blendshape Generator* is a 2D convolutional neural network (CNN), similar to the image translator in Liu et al. [2019], consisting of an identity encoder and a blendshape decoder. The encoder, same as the content encoder in Liu et al. [2019], is made of a few 2D convolutional layers followed by several residual blocks. It takes a neutral expression S_0^j as input and maps it into a content latent code that is a spatial feature map. The decoder consists of several instance normalization residual blocks followed by a couple of up-scale convolutional layers. It decodes the feature vector into the blendshape offsets. To adapt 3D models to a compact representation which is friendly for the 2D CNN, we represent every 3D model as a 2D geometry image by first registering all the input 3D models with a same topology and aligning them in UV space (implementation details in Sec. 7), in which each pixel stores the $x - y - z$ coordinates of one vertex.

Instead of training the generator in one network, we adopt a two-branch architecture inspired by Bai and Ghanem [2017] which uses a multi-branch network for face detection and tracking with different face size.

We observe that the scale of different blendshapes varies greatly. Thus we came up with a two-branch training strategy. We separate our blendshapes into two categories: 14 extreme blendshapes with

relatively large motion and the rest with small motion. As shown in Fig. 5, the two-branch network makes the generated blendshapes more personalized and closer to the reference FACS expression.

4.2 Tuning Stage

In the Estimation Stage, the blending weights are given, and consistent for all subjects, but practically it is hard to guarantee that different subjects can realize the same exact expressions. In this scenario, the fixed blending weights lead to inaccuracy when fitting such expressions for different subjects. Therefore, we relax constraints on the blending weights and instead learn them with a neural network. As shown in Fig. 3, compared to the Estimation Stage, the initial blendshapes work as additional input to the *Blendshape Generator*, and another *Blending Weight Predictor* is introduced to predict blending weights from the input expression in the Tuning Stage.

The *Blending Weight Predictor* shares a similar network architecture as the *Blendshape Generator* which consists of an expression encoder and a blending weight decoder. Given an input expression P_k^j , the encoder maps it to an expression latent code, followed by the decoder which decodes the latent code into a vector of N blending weights whose values are constrained in $[0, 1]$. Combining the blending weights with the personalized blendshapes generated by the *Blendshape Generator*, we reconstruct the input expression using Eq. 1. The loss used to constrain the output of the *Blending Weight Predictor* is the reconstruction loss defined in Eq. 2.

In order to preserve the semantics and personality of the initial blendshapes generated by the Estimation Stage, we define the regularization term as follows:

$$L_{reg_{FT}} = \sum_{i=1}^N \left\| \Delta S_{i_{FT}}^j - \Delta S_i^j \right\|_1, \quad (8)$$

where $\Delta S_{i_{FT}}^j$ are the target blendshape offsets and ΔS_i^j are initial blendshape offsets generated in the Estimation Stage. Thus, the loss function used in the Tuning Stage is:

$$L_{G_{FT}} = L_{rec} + \omega_{reg_{FT}} L_{reg_{FT}}, \quad (9)$$

where $\omega_{reg_{FT}} = 0.1$. In our implementation, we add skip connections from the initial blendshape to the generator output (as shown in the red line in Fig. 3) such that the generator predicts $\Delta S_{i_{FT}}^j - \Delta S_i^j$. Examples of with and without tuning are shown in Fig. 6, we observe that the Tuning Stage achieves better fitting results by fine-tuning the blendshapes, and jointly optimizing blending weights while preserving the semantics and personality.

5 DYNAMIC TEXTURE GENERATION

In this section, we first introduce our compact representation of dynamic texture assets- Compress and Stretch maps, followed by a learning-based method to infer/extract them. Finally, we demonstrate the utilization of our Compress and Stretch maps for rendering at run-time.

5.1 The Representation

Compress and Stretch Maps. When static textures (obtained from a neutral expression) are used to render extensive expressions, the

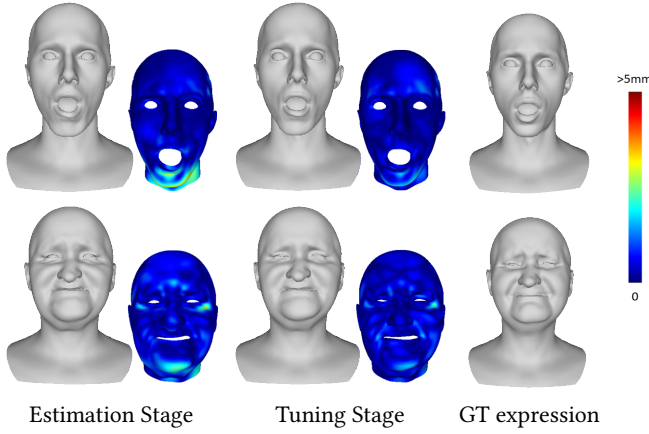


Fig. 6. Comparison of two reconstructed expressions by the Estimation Stage alone and with the addition of the Tuning Stage, along with error maps between the reconstructed expressions and the ground truth expressions. The output from the Tuning Stage results in better reconstruction with smaller fitting errors.

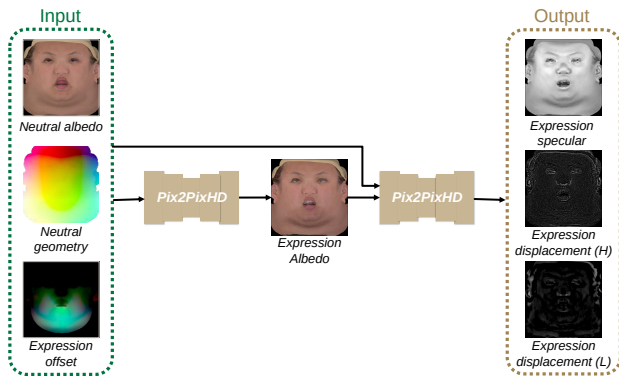


Fig. 7. Texture Generative Network. Given the albedo map and the geometry image of the input model in neutral expression and the geometry image of the target expression offset, the first network generates the albedo map of the expression using pix2pixHD [Wang et al. 2018b]. Then, combining the initial input and predicted albedo map, the second network infers specular intensity, low-frequency, and high-frequency displacement maps.

missing details (e.g. wrinkles) caused by facial motion will significantly reduce the photo-realism of rendering results. Especially for the extreme/exaggerated expressions, high-fidelity muscle movement and micro-expressions make big differences. A natural way to solve this problem is to customize a set of dynamic textures for blendshapes. However, the number of blendshapes used in high-end industries may be of the magnitude of hundreds or thousands. The creation of such large dynamic textures is costly and requires substantial computational power. More importantly, it is difficult to load such a vast collection of dynamic textures into a rendering engine at once, in particular, with multiple layers (e.g. albedo, specular intensity, displacement maps) at high resolution. A memory-efficient, compact, and easy-to-compute dynamic representation is needed.

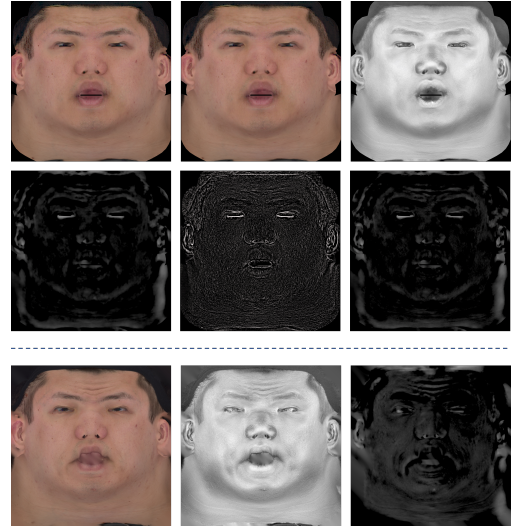


Fig. 8. Generated textures and ground truth textures of an expression. Row 1 from left to right: low-resolution albedo map ($1K \times 1K$), high-resolution albedo map ($4K \times 4K$), and specular intensity map. Row 2 from left to right: low-frequency, high-frequency and combined displacement maps. Row 3 from left to right: ground truth of albedo, specular intensity and displacement maps.



Fig. 9. Illustration of Compress and Stretch Maps. From Top to Down Rows: Neutral Static maps, Compress maps, Stretch maps. From Left to Right Columns: Diffuse Albedo maps, Specular maps, Normals maps (in tangent space) computed from Displacement maps.

Moreover, it should also be expressive enough to cover all the possible dynamic details of facial motion losslessly. We adopt *Compress and Stretch Maps* as shown in Fig. 9 along with a static neutral texture to be the dynamic texture library, which is a commonly adopted

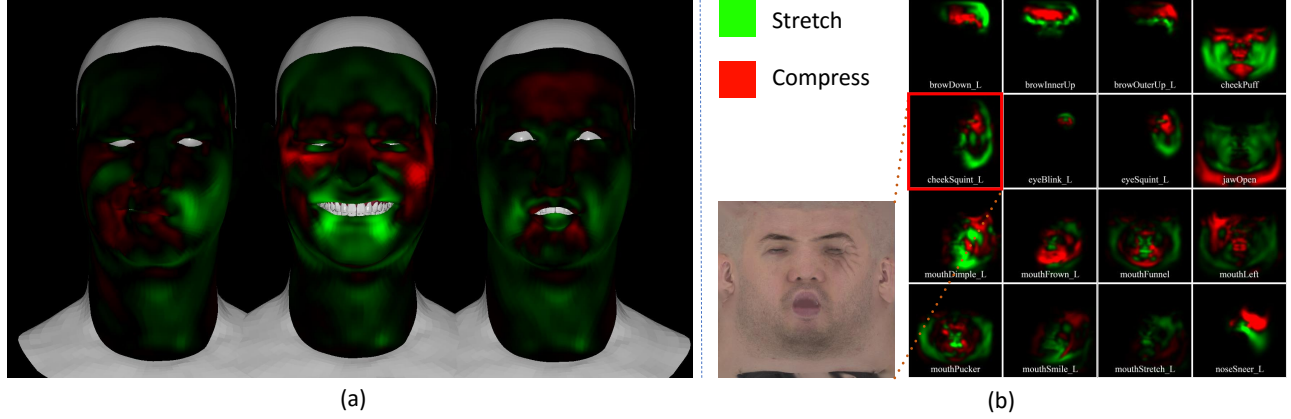


Fig. 10. Illustration of Influence maps. (a). Influence value rendered in geometries with different expressions (*Mouth Right*, *Smile* and *Lip Funnel*). (b). Selected Influence maps from a set of blendshapes and an example of dynamic albedo with its corresponding influence map in the blendshape *CheckSquint_L*. Note that we store compress and stretch influence maps as *R* and *G* channels and set *B* channel to zeros.



Fig. 11. Illustration of Compress Maps Extraction. Left: expression textures generated from networks. Right: compress maps extracted by blending expression textures based on the influence maps. Note that the final compress maps gather all the dynamic details caused by skin local compression (in the orange circles) from all the expressions.

format in the industry [Oat 2007]. Guided by *Influence Maps*, compress and stretch maps gather the most prominent features caused by the local compression/stretching movement of all the available expressions.

Influence Maps. Influence maps are computed based on the geometry changes between the expressions and the neutral face. For each of the vertices x on the neutral mesh N , we define the average edge length of its one-ring neighbors as $E_N(x)$, and then for an arbitrary expression mesh P of the same subject, the influence value of each vertex on P in compress maps can be computed as:

$$I_{P_{Compress}}(x) = \begin{cases} \|E_N(x) - E_P(x)\|, & E_P(x) < E_N(x) \\ 0, & E_P(x) \geq E_N(x) \end{cases} \quad (10)$$

Similarly, the influence value of each vertex on P in stretch maps is as follows:

$$I_{P_{Stretch}}(x) = \begin{cases} \|E_P(x) - E_N(x)\|, & E_P(x) > E_N(x) \\ 0, & E_P(x) \leq E_N(x) \end{cases} \quad (11)$$

Based on the per-vertex influence values, we interpolate a per-pixel compress and stretch influence map as shown in Fig. 10. Note that we store compress and stretch influence maps as *R* and *G* channels separately. The influence maps provide the weights to blend and extract dynamic textures.

5.2 Compress and Stretch Map Generation

In the standard industry pipeline, the compress and stretch maps are handcrafted by skilled artists using numerous captured expressions as reference. To automate this procedure, especially when only a single scan is provided in our scenarios, we came up with a two-step solution. Firstly, we predict the texture maps (*i.e.* albedo, specular intensity, and displacement) of the input subject's pre-defined expressions using a deep neural network. Then a blending step is introduced to fuse them into compress and stretch maps.

Expression Texture Generation Networks. Given a single neutral scan with an albedo map, in order to predict the high-fidelity albedo, specular intensity, and displacement maps of different expressions, we propose a cascade architecture, as shown in Fig. 7. We first take the neutral geometry with its albedo map and the target expression offset from the neutral geometry as input to predict the albedo map offset of the target expression. The predicted offset is then added to the neutral albedo map to generate the expression albedo map as the intermediate results, further combining the input of the first network to be fed into the second network. The second network then infers the specular intensity and displacement maps. Both of the networks are the Pix2pixHD [Wang et al. 2018b] model, which contains an encoder with several CNN layers, followed by a couple of Resnet blocks, and a decoder with similar architecture. The reason of using a cascade network with an expression albedo map as intermediate results include: (1) the specular intensity and displacement maps generated using the albedo map as a prior have fewer artifacts and higher quality; (2) this architecture allows us to handle incomplete training data (some of the subjects do not have the specular intensity and displacement maps). In particular, we separate the displacement map into low-frequency and high-frequency during

training, following Huynh et al. [2018]; Yamaguchi et al. [2018] to make the problem more tractable and merge them together before using. Both input and output of the two networks have $1K \times 1K$ resolution. Furthermore, with all these $1K$ result maps, we up-scale them into $4K \times 4K$ using a pre-trained super-resolution network [Ledig et al. 2017]. In Fig. 8, we show a complete set of expression textures generated by our networks.

Compress and Stretch Map Extraction. We design an algorithm to extract compress and stretch maps based on the influence maps from the above predicted expression textures as shown in Fig. 11. Let I_i be the influence map of the i th expression, and the influence value of each pixel (x, y) is $I_i(x, y)$. We first normalize the influence map of all the expressions with a weighted sum strategy to ensure the spatial consistency among all the expressions as follows (take the compress map as an example):

$$\hat{I}_{iCompress}(x, y) = \frac{e^{I_{iCompress}(x, y)}}{\sum_i e^{I_{iCompress}(x, y)}}, \quad (12)$$

in which $\hat{I}_{iCompress}$ is the normalized influence map of $I_{iCompress}$ ($i = 1 \dots N$) where N is the number of expressions.

Once we get the normalized influence maps, the compress map is computed as follows:

$$T_{Compress}(x, y) = \sum_i \hat{I}_{iCompress}(x, y) T_i(x, y), \quad (13)$$

Where T_i is the texture of the i th expression, and it can be one of the albedo, specular, and displacement. The stretch maps are computed similarly. Finally, we obtain compress and stretch maps for albedo, specular, and displacement maps, respectively.

5.3 Runtime Dynamic Texture Generation

When using dynamic assets for rendering in runtime applications, such as tracking, animation, we first solve the blending weights of each input expression using personalized blendshapes. Those blending weights combined with a set of pre-defined influence maps of blendshapes, will be used to sample the current dynamic textures from compress and stretch maps. The dynamic textures are generated as follows:

$$\begin{aligned} T(x, y) = & T_N(x, y) \\ & + \sum_i^N \left(\alpha_i \hat{I}_{iCompress}(x, y) (T_{Compress}(x, y) - T_N(x, y)) \right. \\ & \left. + \alpha_i \hat{I}_{iStretch}(x, y) (T_{Stretch}(x, y) - T_N(x, y)) \right) \end{aligned} \quad (14)$$

where T_N is the static texture of neutral expression, $T_{Compress}$ and $T_{Stretch}$ correspond to the compress and stretch textures, $\hat{I}_{iCompress}$ and $\hat{I}_{iStretch}$ are the influence maps of the i th blendshape and α_i indicates its blending weight.

6 ASSEMBLY

In addition to the primary dynamic assets (face geometry and textures) generated using networks, we also include secondary components (e.g. eyeballs, lacrimal fluid, eyelashes, teeth, and gums) in our avatar as shown in Fig. 14. We handcrafted a set of generic



Fig. 12. Selected FACS units from Light Stage Dataset. From left to right: Neutral, Eye_close, Lip_corner Puller, Eyes_Up, Lip_Funneler, Inner_Brow_Raiser, Dimpler, Upper_Lip_Raiser, Lower_Lip_Depressor, Outer_Brow_Raiser, Brow_Lowerer, Inner_Brow_Raiser, Lip_Presser.

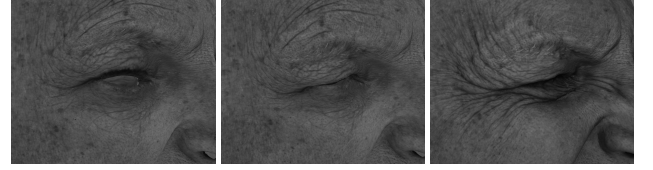


Fig. 13. Laplacian deformation results of neutral mesh to target expression model using (a) landmarks only, and (b) dense optical flow correspondence. (c) Target expression.

blendshapes with all the primary and secondary parts. We further use this set of generic blendshapes to linearly fit each expression generated by our networks based on corresponding vertices on the facial regions. The computed coefficients based on the primary parts drive the secondary components, such that eyelashes will travel with eyelids. The linearly fitted secondary elements will be combined with the primary facial parts to get an integrated face model. Except for eyeball, other secondary parts share a set of generic textures for all the subjects. For eyeball textures, we adopt an eyeball assets database [Kollar 2019] with 90 difference eye textures (pupil patterns) to match with input subjects.

7 DATASET

The facial scan dataset used in training comes from a combined source of aligned face models with $4k$ resolution textures and geometries aligned to a known topology [Li et al. 2020]. The dataset consists of 178 scan subjects divided into two sets, one of 78 (Light Stage), and one of 100 subjects ([Triplegangers 2019]); performing 26 and 20 static FACS expressions respectively. The FACS expressions are fixed, which enables labeling of corresponding weights in our set of template blendshapes. This feature is particularly useful when isolating orthogonal shapes that are combined under the scanning session. One such example may be the combination of *action unit 1* (Inner brow raiser), and *action unit 14* (Dimpler) [Ekman and Friesen 1978]. This makes it possible to significantly reduce the number of scans needed (Fig. 12).

The assumptions that have to be realized under the learning of corresponded face morphologies described in section 3 are (1) a rigid transformation of each subject's skull shape can be found for every expression the subject performs, (2) sparse correspondence among subjects need to be established for a common parameterization to be usable, and (3) dense correspondence among expressions need to be established for each subject to track minute changes in skin deformation using texture maps. Next, we describe how these problems are solved to generate the desired dataset.

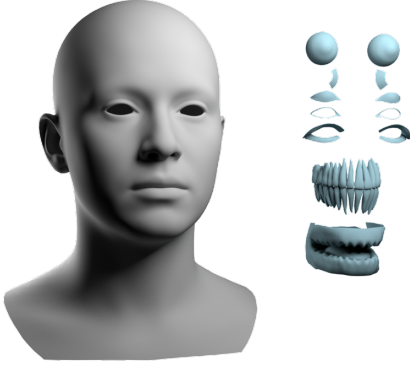


Fig. 14. Our face model consists of multiple parts including face, eyes, eye blend mesh, lacrimal fluid, eye occlusion, eyelashes, teeth, gums and tongue.

7.1 Face Model Registration.

Neutral Scans Registration. First, a linear 3D morphable face (PCA) model is used to fit the neutral face of a scan subject, reconstructed using multi-view stereo [Hsieh et al. 2015]. Secondly, the fitted model is further deformed using the non-rigid iterative closest point method [Li et al. 2008] constrained by facial landmarks [Sagonas et al. 2016]. Additional Laplacian mesh surface warping is applied for surface detail reconstruction [Li et al. 2009].

Expression Scans Registration. We first estimate the blendshape expressions from our template set using the same algorithm, but varying blendshape weights in composite to identity PCA weights and followed by landmarks refinement step. We further introduce a Laplacian deformation step with dense constraints based on multi-view 2D optical flow between the current expression and neutral expression to densely correspond expressions belonging to the same subject [Fyffe et al. 2017], see Fig. 13.

7.2 Texture Data Generation

By leveraging *polarized spherical gradient illumination* [Ghosh et al. 2011; Ma et al. 2007] we can compute skin micro-structure, and material intrinsics such as diffuse albedo and specularity as we have seen inferred by our pixel translation networks. Specifically, these maps are computed on a fixed aligned topology provided by the before mentioned morphable face model.

7.3 Template Blendshape Model

Our blendshape model is based on the naming convention of Apple’s ArKit with additional modifications enabling asymmetries for eye-brow shapes. The shapes were computed by fitting a set of around 50 scanned face neutrals along with their performed FACS shapes. By computing averages over all subjects, keeping each expression fixed, we could find reasonable averages of each shape which could be artistically isolated to keep linear independence and semantic meaning; and to avoid self-intersection.

8 RESULTS

8.1 Implementation Details

We split our data into two subsets: training set (137 subjects) and testing set (41 subjects). Each of the subsets covers a wide span of

Table 1. Run time for each component in our framework.

Component	Time (ms)
Estimation Stage (Single Branch)	2.386
Tuning Stage	2.200
Texture Generation - Albedo map	130.9
Texture Generation - Displacement & Specular	398.1
Texture Generation - Up-scaling	3801

age, gender, and race. We learn our Blendshape generation networks using the RMSProp optimizer with a fixed learning rate of 0.0001 and a batch size of 4. For the texture generation network, it is optimized by the Adam optimizer with a fixed learning rate of 0.0002, batch size of 1. We train Estimation Stage and Tuning Stage for about 50,000 and 60,000 iterations respectively on an NVIDIA GeForce RTX 2080 GPU. And we train texture generation model on NVIDIA Tesla V100.

8.2 Experiments

Run Time. We record the run time of each component for an end-to-end system test (Table 1). Testing of our blendshape generation model was performed on an NVIDIA GeForce RTX 2080 GPU while texture generation was performed on an NVIDIA Tesla V100.

Compared to the standard high resolution avatar generation pipeline, that requires intensive manual work of weeks or months of time along with many reference expressions to be captured, our proposed approach is fast, low-cost, and robust (high-resolution training data ensures the output avatar quality).

Results. In Fig. 15, we show selected expressions of novel subjects rendered using all the assets automatically generated by our framework from different sources of input data. Results show that our generated dynamic textures capture the middle-frequency details such as wrinkles and folds. In particular, the generated blendshapes of different individuals show that our approach captures the user-specific motion properties (e.g. *Mouth Right* in row two, four, six) with the semantics preserved. Note that all the generated subjects are unseen by the networks. Input test data from 3DScanstore [2019] and low-quality data captured by a mobile device are from a different domain and have never been observed by our networks. Hence, these results indicate the robustness of our framework.

Comparison and Evaluation. In Fig. 16, by combining the same neutral with the corresponding personalized Blendshapes units (*Jaw Open* and *Mouth Right*) belonging to different individuals, we showcase that our network is successful in imposing user-specific motion features to the template blendshapes.

In Fig. 17, we show an extreme expression’s fitting results with template blendshapes and our generated personalized blendshapes separately. Results indicate that our generated personalized blendshapes perform better in the non-rigid deformation (e.g. double-chin when open mouth).

In Fig. 18, we demonstrate the influence of personalized blendshapes on reconstruction/tracking accuracy by swapping blendshapes of two subjects during expression tracking. Results show that personalized blendshapes will be more expressive to the input

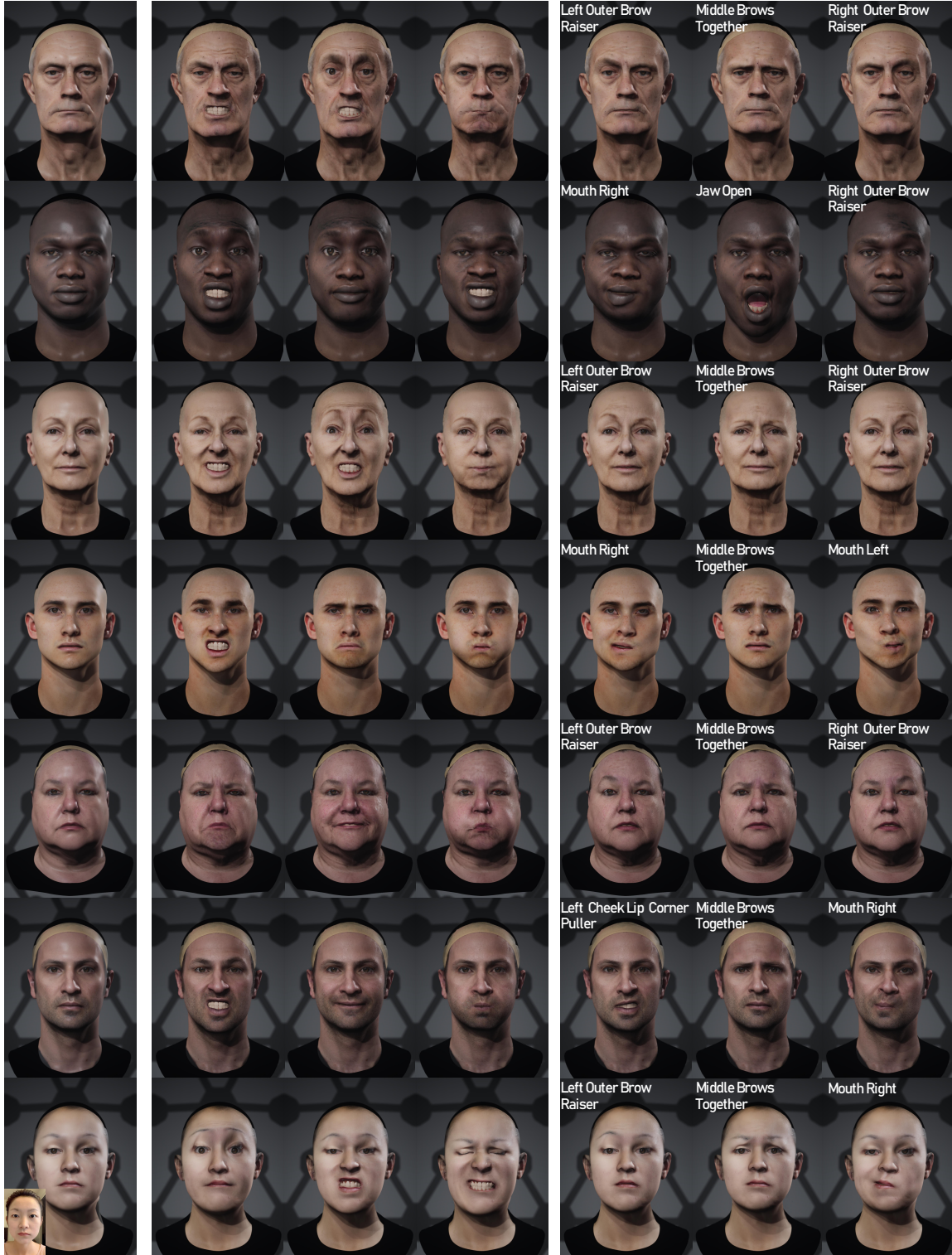


Fig. 15. Expressions reconstructed by face rig assets generated by our framework with inputs from multiple sources. From left to right: Column 1: input neutral including geometry and albedo. Column 2 to Column 4: selected reconstructed expressions. Column 5 to Column 7: selected blendshape units. From top to bottom: Row 1 and Row 2: input neutral from Triplegangers [Triplegangers 2019], Row 3 and Row 4: input neutral from online resources [3DScanstore 2019], Row 5 and Row 6: input neutral from Light Stage testing set. Row 7: Input neutral from iPhone X Arkit. The last example shows that our method can also be applied to data captured by a low-quality device despite that low-resolution input image may reduce the resulting quality.

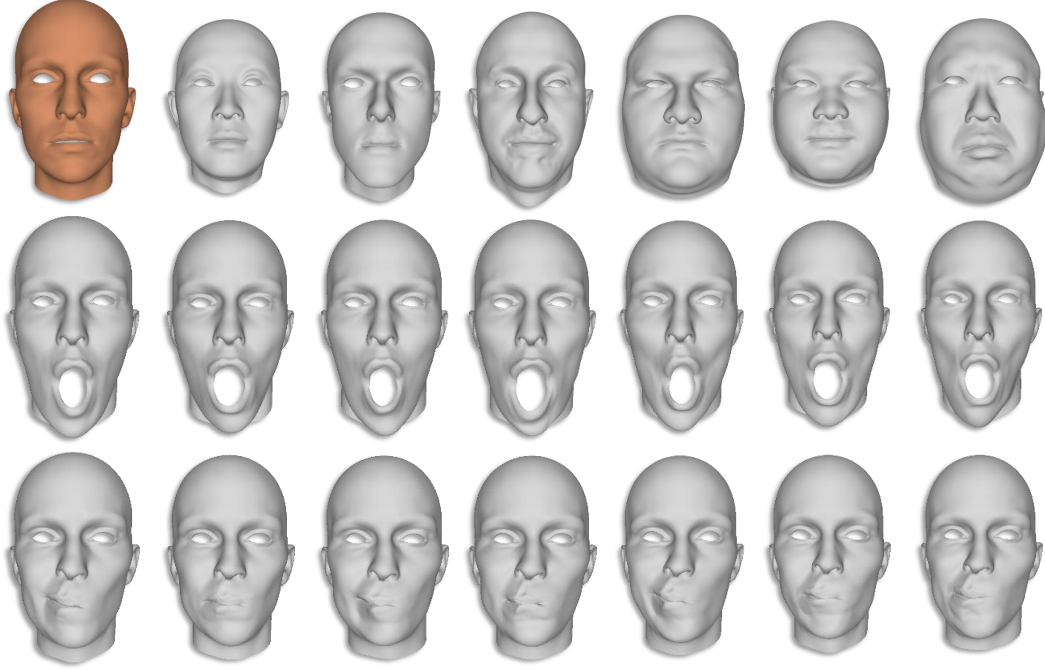


Fig. 16. Demonstration of customized identity of individuals on our generated blendshapes expressions. We combine blendshapes units from different individuals with the same template neutral (shown in orange). Row 1: source individuals. Row 2: combine personalized *Jaw_Open* of individuals in row one with template neutral. Row 3: combine personalized *Mouth_Right* of individuals in row one with template neutral.

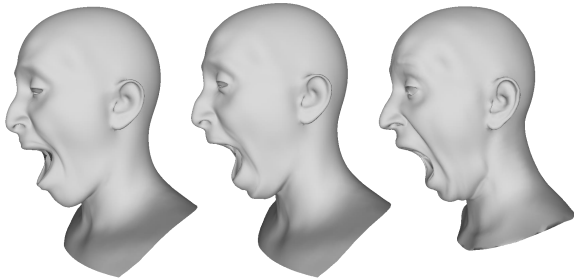


Fig. 17. Comparison of extreme expression fitting using template blendshapes and our generated personalized blendshapes. Left: fitting results using template blendshapes. Middle: fitting results using our generated Personalized blendshapes. Right: ground truth expression.

identity regarding tracking accuracy, especially in the facial part with more non-linear and large motion (e.g. Mouth). This result also demonstrates the effectiveness of our network: One of our network objective is to achieve better reconstruction of scanned expression.

In Fig. 19, we further compare our generated blendshapes with template blendshapes and the method of Li et al. [2010]. Results show that our approach is comparable to Li et al. [2010] in the task of imposing personality to template blendshapes. Note that in Li et al. [2010], 26 references scanned expression are used for optimization purposes. On the other side, our results are obtained based on a

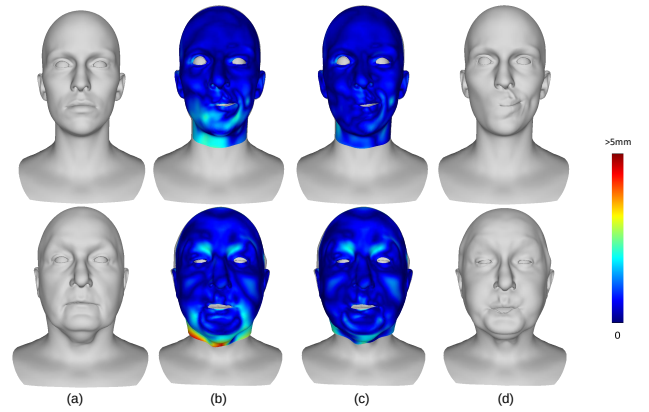


Fig. 18. Numerical analysis of the expressiveness of personalized blendshapes on expression tracking by swapping Blendshapes. (a) Neutrals of two individuals. (b) Reconstruction error using personalized Blendshapes from counterpart individuals. (c) Reconstruction error using their own personalized Blendshapes. (d) Target expressions.

single neutral scan. Another observation is that our deep learning-based method shows more robust results with fewer artifacts (e.g. the left mouth corner on the blendshape *Mouth Left*).

In Fig. 20, we show dynamic displacement generated by our framework on novel subjects. Results show the effectiveness of

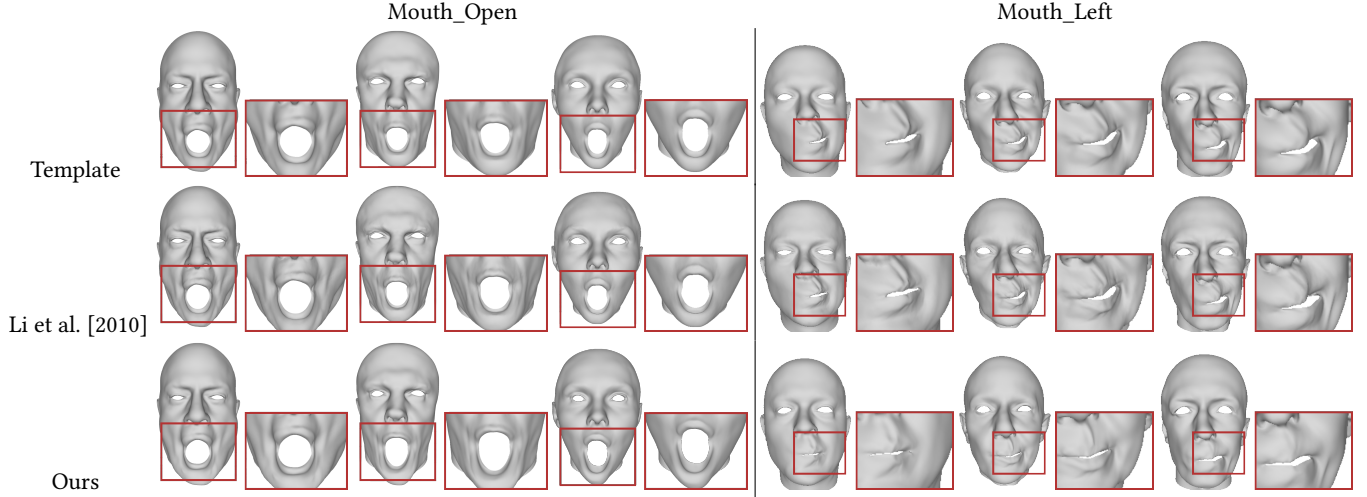


Fig. 19. Comparison of selected generated Blendshapes units with template generic and Li et al. [2010]. Row 1: template blendshapes generated by expression transferred from a set of generic blendshapes using method in Sumner and Popović [2004]. Row 2: blendshapes optimized with method in Li et al. [2010]. Row 3: our method. Note that the results generated in Li et al. [2010] are from 26 scanned expressions, ours are from a single neutral input.

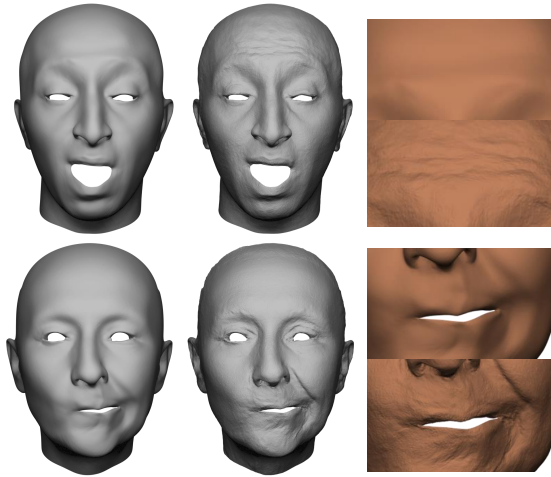


Fig. 20. Dynamic displacement map predicted by our framework. Left: base geometries. Middle: results by applying generated displacement to base geometries. Right: closed-up comparison before and after applying dynamic displacement maps.

our displacement network, which infers middle frequency details (e.g. wrinkles) as well as high-frequency mesoscopic details.

In Fig. 23, we show the results and comparison of our generated dynamic textures on different subjects. Compared to static albedo from input neutral, our generated dynamic albedo predicts wrinkles, and folds caused by local self-occlusion of middle-frequency geometry change during deformation. The results also show that our predicted dynamic specular and displacement maps add mesoscopic details on top of diffuse albedo. It greatly improves the visual realism of rendering, which is important for high-end applications.

Table 2. Reconstruction errors between the ground truth expressions and the reconstructed expressions using blendshapes by different methods on training and testing datasets.

Method	Training ↓	Testing ↓
Template blendshapes	1.661	1.638
Optimization method [Li et al. 2010]	1.389	1.483
Ours	1.341	1.372

In Fig. 21, we compare our generated full set of face rig assets with the state-of-the-art paGAN [Nagano et al. 2018]. Note the the base geometry used by paGAN [Nagano et al. 2018] are reconstructed from a single frontal image while ours are based on a high-quality scan. Compared to paGAN, our avatar shows better quality and much more details, which indicates that a good quality neutral scan serves better in the task of high-end avatar generation. The results also shows the unique physically-based skin assets will greatly improve the avatar rendering quality. The displacement map in our assets captures the middle frequency and pore-level details.

8.3 Applications

Expression Reconstruction/ Face Tracking. In Fig. 22, we compare our generated personalized blendshapes on fitting of performance capture sequences with other methods. As shown in Fig. 18, smaller fitting errors indicates better personality on blendshapes. Results show that our generated personalized blendshapes outperform baseline methods (Template and optimization-based method in Li et al. [2010] on accuracy of the face tracking task using the same solver. To provide better quantitative evidence, we evaluate face reconstruction on 2,548 expressions in training dataset and 626 expressions in testing datasets. The results are listed in Table 2. Blendshapes optimized by Li et al. [2010] and ours show smaller reconstruction errors in both training and testing data.

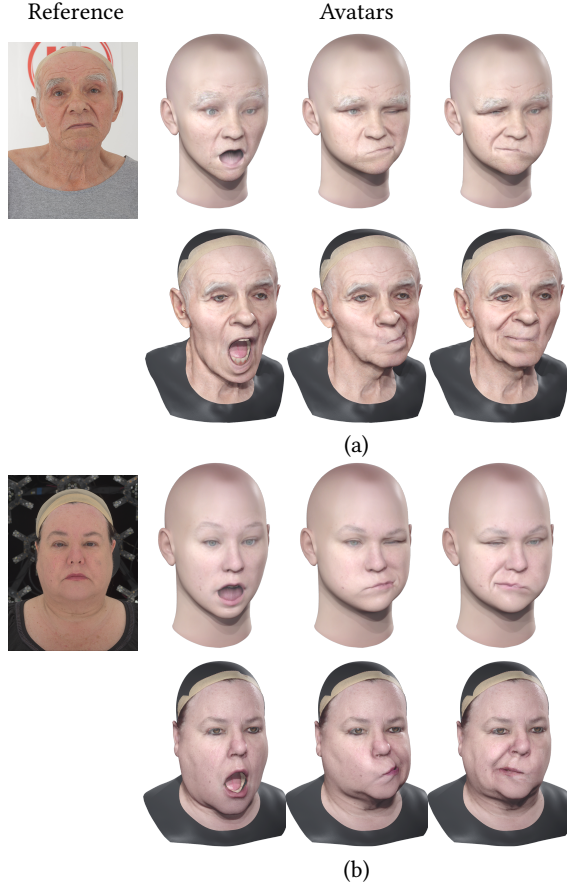


Fig. 21. Comparison of generated face avatars between paGAN [Nagano et al. 2018] and our method. (a) and (b) show two cases of generated avatars from the neutral model in reference images. In each case, Row 1 shows the avatars generated by paGAN [Nagano et al. 2018] while Row 2 shows our results.

Animation. In Fig. 24, we show that our generated face rig assets can be used directly for animation. *Please refer to accompanying video material for more results.*

9 CONCLUSION

We have demonstrated an end-to-end framework for high-quality personalized face rig and asset generation from a single scan. Our face rig assets include a set of personalized blendshapes, physically based dynamic textures and secondary facial components (including teeth, eyeballs, and eyelashes). Compared to previous automatic avatar and facial rig generation approaches, which either require a considerable number of person-specific scans or can only produce a relatively low-fidelity avatar, our framework only requires a single neutral scan as input and can produce plausible identity attributes including physically-based dynamic textures of facial skins. This characteristic is key to creating compelling animation-ready avatars at scale.

We achieve the above objective by modeling the correlation between identity and personalized blendshapes using an extensive dataset of high-resolution facial scans. In particular, our generated dynamic textures add details from mid-frequencies (wrinkles) to mesoscopic ones (pore level). Our automatically generated face rig assets are valuable for real-world production pipelines, as these high-fidelity initial models can be provided to artists for fine-tuning or simply used as secondary characters for crowds. Our proposed method is fast, robust, and lightweight, allowing production studios to simply scan a neutral face of a person and immediately obtain a high-quality facial rig. An interesting insight from our experiments is that the identity seems to be enough for a plausible inference of personalized facial appearance and dynamic expressions. In addition to our framework, we have also introduced a novel self-supervised deep neural network training approach to deal with the case when no ground truth data is available, which in our case are the personalized blendshapes.

Limitations and Future Work. As a deep learning approach, the effectiveness of our algorithm relies on the variety and volume of training data of our database. In particular, facial expressions that are specific to young subjects could be improved, due to the lack of young subjects in our current database. For the same reason, our framework also does not perform well on subjects with facial hair or beard as shown in Fig. 25. We plan to augment our database to cover more diversity and appearance variations.

Our template model consists of 55 blendshape vectors, which can recover most of the expressions in daily life and is commonly used in lightweight applications. However, certain extreme expressions still cannot be represented by our model. Our proposed network architecture can be adapted for arbitrary template blendshapes. Thus, we are interested in exploring more sophisticated blendshape rigs that consist of hundreds to thousands of expressions, such as the ones used in film production. We use generic eyes and teeth models for all the generated avatars. An interesting direction would be to explore how to generate personalized eyes [Bérard et al. 2016, 2019] and teeth [Velinov et al. 2018; Wu et al. 2016] automatically as well.

ACKNOWLEDGMENTS

We thank Liwen Hu from Pinscreen for the fruitful discussions and helping with this paper. This research is funded by in part by the ONR YIP grant N00014-17-S-FO14, the CONIX Research Center, one of six centers in JUMP, a Semiconductor Research Corporation (SRC) program sponsored by DARPA, the Andrew and Erna Viterbi Early Career Chair, the U.S. Army Research Laboratory (ARL) under contract number W911NF-14-D-0005, Adobe, and Sony. The content of the information does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

REFERENCES

- 3DScanstore. 2019. 3D Scan Store: Male and female 3d head model 48 x bundle. <http://precog.iitd.edu.in/people/anupama>. Online; Accessed: 2019-12-20.
- Victoria Fernández Abrevaya, Adnane Boukhayma, Stefanie Wuhler, and Edmond Boyer. 2019. A Decoupled 3D Facial Shape Model by Adversarial Training. In *CVPR*.

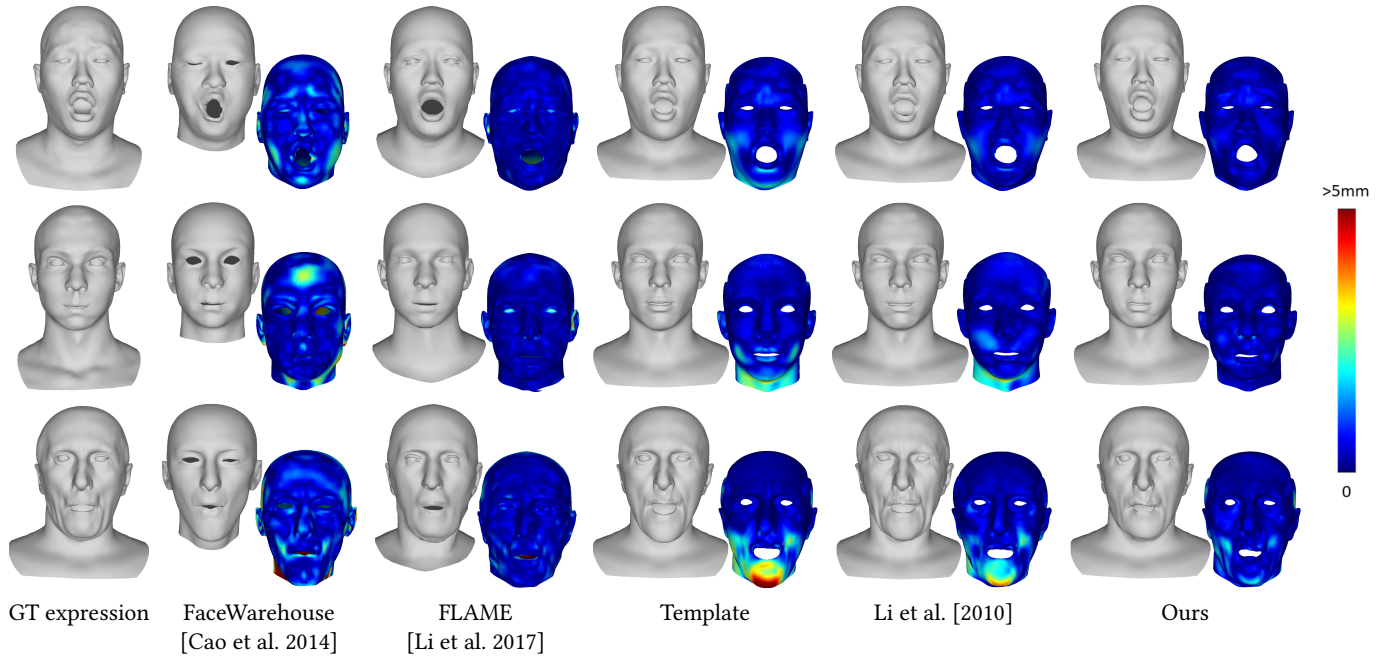


Fig. 22. Comparison on the task of face fitting using different methods.

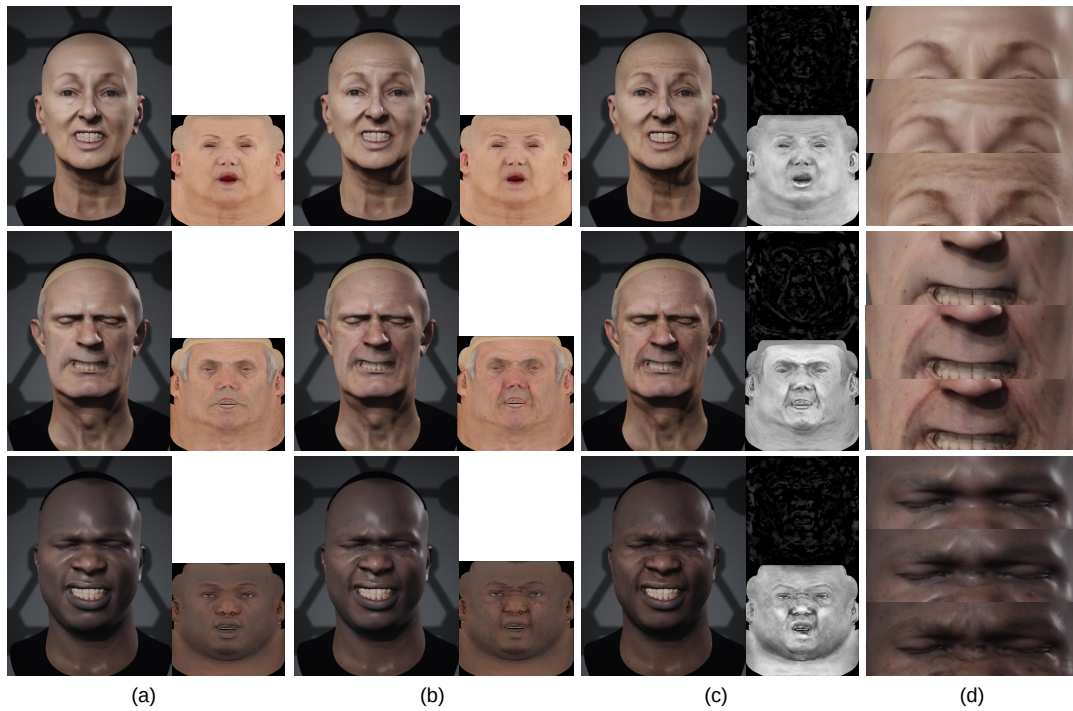


Fig. 23. Results and comparison on Dynamic Textures. (a) Input static albedo and expression renders. (b) Our generated dynamic albedo for the specific expression and renders. (c) Our generated dynamic specular and displacement maps and renders using full set of generated assets (dynamic albedo, specular intensity and displacement). (d) From top to bottom: close-up of skin details of (a), (b) and (c).



Fig. 24. Animation sequences using the full set of our generated assets. Row 1: source sequences. Row 2 to Row 3: target sequences driven by source sequences.

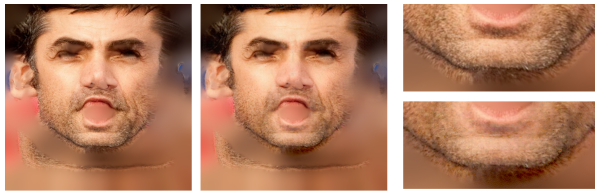


Fig. 25. A failure case of our texture generation model. In this case, we first extract albedo map from an image (taken from CelebA Dataset [Liu et al. 2015]), then feed this map to our texture generation network. From left to right columns: input neutral static albedo map; generated dynamic albedo of one expression by our network; close-up details of static (Top) and dynamic (Down) albedo. Note that our result has slight distortion and miscoloration in some area. This is mainly due to the limited quality of the input image, baked-in lighting and the individual's beard, which our network has not learned how to handle during the training.

Oleg Alexander, Mike Rogers, William Lambeth, Matt Chiang, and Paul Debevec. 2009. The Digital Emily Project: Photoreal Facial Modeling and Animation. In *ACM SIGGRAPH 2009 Courses (SIGGRAPH '09)*. Article Article 12.

Brian Amberg, Reinhard Kothke, and Thomas Vetter. 2008. Expression Invariant 3D Face Recognition with a Morphable Model. In *International Conference on Automatic Face Gesture Recognition*. 1–6.

Timur Bagautdinov, Chenglei Wu, Jason Saragih, Pascal Fua, and Yaser Sheikh. 2018. Modeling Facial Geometry Using Compositional VAEs. In *CVPR*.

Yancheng Bai and Bernard Ghanem. 2017. Multi-branch fully convolutional network for face detection. *arXiv preprint arXiv:1707.06330* (2017).

Thabo Beeler, Bernd Bickel, Paul Beardsley, Bob Sumner, and Markus Gross. 2010. High-Quality Single-Shot Capture of Facial Geometry. *ACM Trans. Graph.* 29, 4, Article Article 40 (2010).

Thabo Beeler, Fabian Hahn, Derek Bradley, Bernd Bickel, Paul Beardsley, Craig Gotsman, Robert W. Sumner, and Markus Gross. 2011. High-Quality Passive Facial Performance Capture Using Anchor Frames. *ACM Trans. Graph.* 30, 4, Article Article 75 (2011).

Pascal Bérard, Derek Bradley, Markus Gross, and Thabo Beeler. 2016. Lightweight Eye Capture Using a Parametric Model. *ACM Trans. Graph.* 35, 4, Article Article 117 (2016).

Pascal Bérard, Derek Bradley, Markus Gross, and Thabo Beeler. 2019. Practical Person-Specific Eye Rigging. *Computer Graphics Forum* (2019). <https://doi.org/10.1111/cgf.13650>

Volker Blanz and Thomas Vetter. 1999. A Morphable Model for the Synthesis of 3D Faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '99)*. ACM Press/Addison-Wesley Publishing Co., USA, 187–194. <https://doi.org/10.1145/311535.311556>

James Booth, Epameinondas Antonakos, Stylianos Ploumpis, George Trigeorgis, Yannis Panagakis, and Stefanos Zafeiriou. 2017. 3D Face Morphable Models "In-the-Wild". In *CVPR*.

James Booth, Anastasios Roussos, Stefanos Zafeiriou, Allan Ponniah, and David Dunaway. 2016. A 3d morphable model learnt from 10,000 faces. In *CVPR*.

Sofien Bouaziz, Yangang Wang, and Mark Pauly. 2013. Online Modeling for Realtime Facial Animation. *ACM Trans. Graph.* 32, 4, Article Article 40 (2013).

Derek Bradley, Wolfgang Heidrich, Tiberiu Popa, and Alla Sheffer. 2010. High Resolution Passive Facial Performance Capture. *ACM Trans. Graph.* 29, 4, Article Article 41 (2010), 10 pages.

Chen Cao, Yanlin Weng, Shun Zhou, Yiyang Tong, and Kun Zhou. 2014. FaceWarehouse: A 3D Facial Expression Database for Visual Computing. *IEEE Transactions on Visualization and Computer Graphics* 20, 3 (2014).

Chen Cao, Hongzhi Wu, Yanlin Weng, Tianjia Shao, and Kun Zhou. 2016. Real-Time Facial Animation with Image-Based Dynamic Avatars. *ACM Trans. Graph.* 35, 4, Article Article 126 (2016).

E Carrigan, E Zell, C Guiard, and R McDonnell. 2020. Expression Packing: As-Few-As-Possible Training Expressions for Blendshape Transfer. In *Computer Graphics Forum*, Vol. 39. Wiley Online Library, 219–233.

Dan Casas, Andrew Feng, Oleg Alexander, Graham Fyffe, Paul Debevec, Ryosuke Ichikari, Hao Li, Kyle Olszewski, Evan Suma, and Ari Shapiro. 2016. Rapid Photorealistic Blendshape Modeling from RGB-D Sensors. In *CASA*.

- Anpei Chen, Zhang Chen, Guli Zhang, Kenny Mitchell, and Jingyi Yu. 2019. Photo-Realistic Facial Details Synthesis From Single Image. In *ICCV*.
- Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. 2018. Grad-Norm: Gradient Normalization for Adaptive Loss Balancing in Deep Multitask Networks. In *ICML*.
- Shiyang Cheng, Michael M. Bronstein, Yuxiang Zhou, Irene Kotsia, Maja Pantic, and Stefanos Zafeiriou. 2019. MeshGAN: Non-linear 3D Morphable Models of Faces. *CoRR* (2019). <http://arxiv.org/abs/1903.10384>
- Paul Ekman and Wallace V. Friesen. 1978. Facial action coding system: a technique for the measurement of facial movement. In *Consulting Psychologists Press*.
- G. Fyffe, P. Graham, B. Tunwattanapong, A. Ghosh, and P. Debevec. 2016. Near-Instant Capture of High-Resolution Facial Geometry and Reflectance. In *Proceedings of the 37th Annual Conference of the European Association for Computer Graphics (EG '16)*. Eurographics Association, Goslar, DEU, 353–363.
- Graham Fyffe, Andrew Jones, Oleg Alexander, Ryosuke Ichikari, and Paul Debevec. 2014. Driving high-resolution facial scans with video performance capture. *ACM Trans. Graph.* 34, 1 (2014), 1–14.
- Graham Fyffe, Koki Nagano, Loc Huynh, Shunsuke Saito, Jay Busch, Andrew Jones, Hao Li, and Paul Debevec. 2017. Multi-View Stereo on Consistent Face Topology. In *Computer Graphics Forum*, Vol. 36. Wiley Online Library, 295–309.
- Pablo Garrido, Michael Zollhöfer, Dan Casas, Levi Valgaerts, Kiran Varanasi, Patrick Pérez, and Christian Theobalt. 2016. Reconstruction of Personalized 3D Face Rigs from Monocular Video. *ACM Trans. Graph.* 35, 3, Article Article 28 (2016), 15 pages.
- Abhijeet Ghosh, Graham Fyffe, Borom Tunwattanapong, Jay Busch, Xueming Yu, and Paul Debevec. 2011. Multiview face capture using polarized spherical gradient illumination. *ACM Trans. Graph.* 30, 6, 129.
- Paulo Gotardo, Jérémy Riviere, Derek Bradley, Abhijeet Ghosh, and Thabo Beeler. 2018. Practical Dynamic Facial Appearance Modeling and Acquisition. *ACM Trans. Graph.* 37, 6, Article Article 232 (2018), 13 pages.
- Pei-Lun Hsieh, Chongyang Ma, Jihun Yu, and Hao Li. 2015. Unconstrained realtime facial performance capture. *CVPR* (2015).
- Liwen Hu, Shunsuke Saito, Lingyu Wei, Koki Nagano, Jaewoo Seo, Jens Fursund, Iman Sadeghi, Carrie Sun, Yen-Chun Chen, and Hao Li. 2017. Avatar Digitization from a Single Image for Real-Time Rendering. *ACM Trans. Graph.* 36, 6, Article Article 195 (2017).
- Haoda Huang, Jinxiang Chai, Xin Tong, and Hsiang-Tao Wu. 2011. Leveraging Motion Capture and 3D Scanning for High-Fidelity Facial Performance Acquisition. *ACM Trans. Graph.* 30, 4, Article Article 74 (2011).
- Loc Huynh, Weikai Chen, Shunsuke Saito, Jun Xing, Koki Nagano, Andrew Jones, Paul Debevec, and Hao Li. 2018. Mesoscopic Facial Geometry Inference Using Deep Neural Networks. In *CVPR*.
- Alexandru Eugen Ichim, Sofien Bouaziz, and Mark Pauly. 2015. Dynamic 3D avatar creation from hand-held video input. *ACM Trans. Graph.* 34, 4 (2015), 1–14.
- Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. 2017. Image-to-Image Translation with Conditional Adversarial Networks. In *CVPR*.
- Ira Kemelmacher-Shlizerman. 2013. Internet-based Morphable Model. *ICCV* (2013).
- Andor Kollar. 2019. Realistic Human Eye. <http://kollarandor.com/gallery/3d-human-eye/>. Online; Accessed: 2019-7-30.
- Samuli Laine, Tero Karras, Timo Aila, Antti Herva, Shunsuke Saito, Ronald Yu, Hao Li, and Jaakko Lehtinen. 2017. Production-level facial performance capture using deep convolutional neural networks. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 1–10.
- Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. 2017. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*.
- Jessica Lee, Deva Ramanan, and Rohit Girdhar. 2019. MetaPix: Few-Shot Video Retargeting. [arXiv:cs.CV/1910.04742](https://arxiv.org/abs/1910.04742)
- J. P. Lewis, Ken Anjyo, Taehyun Rhee, Mengjie Zhang, Fred Pighin, and Zhigang Deng. 2014. Practice and Theory of Blendshape Facial Models. In *Eurographics 2014 - State of the Art Reports*. The Eurographics Association.
- Hao Li, Bart Adams, Leonidas J Guibas, and Mark Pauly. 2009. Robust single-view geometry and motion reconstruction. *ACM Trans. Graph.* 28, 5 (2009), 1–10.
- Hao Li, Robert W. Sumner, and Mark Pauly. 2008. Global Correspondence Optimization for Non-rigid Registration of Depth Scans. In *Proceedings of the Symposium on Geometry Processing (SGP '08)*. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 1421–1430. <http://dl.acm.org/citation.cfm?id=1731309.1731326>
- Hao Li, Thibaut Weise, and Mark Pauly. 2010. Example-Based Facial Riggering. *ACM Trans. Graph.*, Article Article 32 (2010).
- Hao Li, Jihun Yu, Yuting Ye, and Chris Bregler. 2013. Realtime Facial Animation with On-the-Fly Correctives. *ACM Trans. Graph.* 32, 4, Article Article 42 (2013).
- Ruilong Li, Karl Bladin, Yajie Zhao, Chinmay Chinara, Owen Ingraham, Pengda Xi, Xinglei Ren, Pratusha Prasad, Bipin Kishore, Jun Xing, et al. 2020. Learning Formation of Physically-Based Face Attributes. In *CVPR*.
- Tianye Li, Timo Bolkart, Michael J. Black, Hao Li, and Javier Romero. 2017. Learning a Model of Facial Shape and Expression from 4D Scans. *ACM Trans. Graph.* 36, 6, Article Article 194 (2017).
- Or Litany, Alex Bronstein, Michael Bronstein, and Ameesh Makadia. 2018. Deformable Shape Completion with Graph Convolutional Autoencoders. In *CVPR*.
- Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. 2019. Few-shot unsupervised image-to-image translation. In *ICCV*.
- Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2015. Deep Learning Face Attributes in the Wild. In *ICCV*.
- Stephen Lombardi, Jason Saragih, Tomas Simon, and Yaser Sheikh. 2018. Deep appearance models for face rendering. *ACM Trans. Graph.* 37, 4 (2018), 68.
- Wan-Chun Ma, Tim Hawkins, Pieter Peers, Charles-Felix Chabert, Malte Weiss, and Paul Debevec. 2007. Rapid Acquisition of Specular and Diffuse Normal Maps from Polarized Spherical Gradient Illumination. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques (EGSR'07)*. Eurographics Association, Goslar, DEU, 183–194.
- Wan-Chun Ma, Mathieu Lamarre, Etienne Danvoye, Chongyang Ma, Manny Ko, Javier von der Pahlen, and Cyrus A Wilson. 2016. Semantically-aware blendshape rigs from facial performance measurements. In *SIGGRAPH ASIA 2016 Technical Briefs*. ACM, 3.
- Ron Kimmel Matan Sela, Elad Richardson. 2017. Unrestricted Facial Geometry Reconstruction Using Image-to-Image Translation. In *ICCV*.
- Arnold Maya. 2019. Maya Arnold renderer. <https://arnoldrenderer.com/>. Online; Accessed: 2019-11-22.
- Koki Nagano, Jaewoo Seo, Jun Xing, Lingyu Wei, Zimo Li, Shunsuke Saito, Aviral Agarwal, Jens Fursund, and Hao Li. 2018. PaGAN: Real-Time Avatars Using Dynamic Textures. *ACM Trans. Graph.* 37, 6, Article Article 258 (2018), 12 pages.
- Jun-yong Noh and Ulrich Neumann. 2001. Expression Cloning. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '01)*.
- Christopher Oat. 2007. Animated wrinkle maps. In *ACM SIGGRAPH 2007 courses*. 33–37.
- Kyle Olszewski, Joseph J. Lim, Shunsuke Saito, and Hao Li. 2016. High-Fidelity Facial and Speech Animation for VR HMDs. *ACM Trans. Graph.* 35, 6 (2016).
- Hayato Onizuka, Diego Thomas, Hideaki Uchiyama, and Rin-ichiro Taniguchi. 2019. Landmark-Guided Deformation Transfer of Template Facial Expressions for Automatic Generation of Avatar Blendshapes. In *ICCVW*.
- Chandan Pawaskar, Wan-Chun Ma, Kieran Carnegie, John P Lewis, and Taehyun Rhee. 2013. Expression transfer: A system to build 3D blend shapes for facial animation. In *2013 28th International Conference on Image and Vision Computing New Zealand (IVCNZ 2013)*. IEEE, 154–159.
- Anurag Ranjan, Timo Bolkart, Soubhik Sanyal, and Michael J. Black. 2018. Generating 3D Faces Using Convolutional Mesh Autoencoders. In *ECCV*.
- Christos Sagonas, Epameinondas Antonakos, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. 2016. 300 faces in-the-wild challenge: Database and results. *Image and vision computing* 47 (2016), 3–18.
- Robert W. Sumner and Jovan Popović. 2004. Deformation Transfer for Triangle Meshes. *ACM Trans. Graph.* 23, 3 (2004), 399–405.
- Ayush Tewari, Michael Zollhofer, Hyeonwoo Kim, Pablo Garrido, Florian Bernard, Patrick Perez, and Christian Theobalt. 2017. MoFA: Model-Based Deep Convolutional Face Autoencoder for Unsupervised Monocular Reconstruction. In *ICCV*.
- Justus Thies, Michael Zollhöfer, Matthias Nießner, Levi Valgaerts, Marc Stamminger, and Christian Theobalt. 2015. Real-Time Expression Transfer for Facial Reenactment. *ACM Trans. Graph.* 34, 6 (2015).
- Justus Thies, Michael Zollhöfer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. 2016. Face2Face: Real-Time Face Capture and Reenactment of RGB Videos. In *CVPR*.
- Luan Tran, Feng Liu, and Xiaoming Liu. 2019. Towards High-fidelity Nonlinear 3D Face Morphable Model. In *CVPR*.
- Luan Tran and Xiaoming Liu. 2018. Nonlinear 3D Face Morphable Mmodel. In *CVPR*.
- Triplegangers. 2019. Triplegangers Face Models. <https://triplegangers.com/>. Online; Accessed: 2019-12-21.
- Levi Valgaerts, Chenglei Wu, Andrés Bruhn, Hans-Peter Seidel, and Christian Theobalt. 2012. Lightweight Binocular Facial Performance Capture under Uncontrolled Lighting. *ACM Trans. Graph.* 31, 6 (2012).
- Zdravko Velinov, Marios Papas, Derek Bradley, Paulo Gotardo, Parsa Mirdehghan, Steve Marschner, Jan Novák, and Thabo Beeler. 2018. Appearance Capture and Modeling of Human Teeth. *ACM Trans. Graph.* 37, 6, Article Article 207 (Dec. 2018).
- Daniel Vlasic, Matthew Brand, Hanspeter Pfister, and Jovan Popović. 2005. Face Transfer with Multilinear Models. *ACM Trans. Graph.* 24, 3 (2005), 426–433.
- Ting-Chun Wang, Ming-Yu Liu, Andrew Tao, Guilin Liu, Jan Kautz, and Bryan Catanzaro. 2019. Few-shot Video-to-Video Synthesis. In *NeurIPS*.
- Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. 2018a. Video-to-Video Synthesis. In *NeurIPS*.
- Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. 2018b. High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs. In *CVPR*.

- Shih-En Wei, Jason Saragih, Tomas Simon, Adam W Harley, Stephen Lombardi, Michal Perdoch, Alexander Hypes, Dawei Wang, Hernan Badino, and Yaser Sheikh. 2019. VR facial animation via multiview image translation. *ACM Trans. Graph.* 38, 4 (2019), 1–16.
- Thibaut Weise, Hao Li, Luc Van Gool, and Mark Pauly. 2009. Face/Off: live facial puppetry. In *SCA '09*.
- Chenglei Wu, Derek Bradley, Pablo Garrido, Michael Zollhöfer, Christian Theobalt, Markus Gross, and Thabo Beeler. 2016. Model-Based Teeth Reconstruction. *ACM Trans. Graph.* 35, 6, Article Article 220 (2016).
- Shugo Yamaguchi, Shunsuke Saito, Koki Nagano, Yajie Zhao, Weikai Chen, Kyle Olszewski, Shigeo Morishima, and Hao Li. 2018. High-fidelity facial reflectance and geometry inference from an unconstrained image. *ACM Trans. Graph.* 37, 4 (2018), 1–14.
- Haotian Yang, Hao Zhu, Yanru Wang, Mingkai Huang, Qiu Shen, Ruigang Yang, and Xun Cao. 2020. FaceScape: a Large-scale High Quality 3D Face Dataset and Detailed Riggable 3D Face Prediction. In *CVPR*.
- Li Zhang, Noah Snavely, Brian Curless, and Steven M. Seitz. 2004. Spacetime Faces: High Resolution Capture for Modeling and Animation. *ACM Trans. Graph.* 23, 3 (2004).
- Yuxiang Zhou, Jiankang Deng, Irene Kotsia, and Stefanos Zafeiriou. 2019. Dense 3D Face Decoding Over 2500FPS: Joint Texture & Shape Convolutional Mesh Decoders. In *CVPR*.