paGAN: Real-time Avatars Using Dynamic Textures

KOKI NAGANO, Pinscreen, USC Institute for Creative Technologies
JAEWOO SEO, Pinscreen
JUN XING, USC Institute for Creative Technologies
LINGYU WEI, Pinscreen
ZIMO LI, University of Southern California
SHUNSUKE SAITO, University of Southern California, Pinscreen
AVIRAL AGARWAL, Pinscreen
JENS FURSUND, Pinscreen
HAO LI, Pinscreen, University of Southern California, USC Institute for Creative Technologies

Fig. 1. Overview of our system. Using a single neutral-face input image, we are able to synthesize arbitrary expressions both in image space and UV texture space. These generated textures, which include a photoreal mouth interior and the eyes, can then be used to pilot dynamic avatars in real-time with minimal computational resources, usable even in a mobile environment.

With the rising interest in personalized VR and gaming experiences comes the need to create high quality 3D avatars that are both low-cost and variegated. Due to this, building dynamic avatars from a single unconstrained input image is becoming a popular application. While previous techniques that attempt this require multiple input images or rely on transferring dynamic facial appearance from a source actor, we are able to do so using only one 2D input image without any form of transfer from a source image. We achieve this using a new conditional Generative Adversarial Network design that allows fine-scale manipulation of any facial input image into a new expression while preserving its identity. Our photoreal avatar GAN (paGAN) can also synthesize the unseen mouth interior and control the eye-gaze direction of the output, as well as produce the final image from a novel viewpoint. The method is even capable of generating fully-controllable temporally stable video sequences, despite not using temporal information during training. After training, we can use our network to produce dynamic image-based avatars that are controllable on mobile devices in real time. To do this, we compute a fixed set of output images that correspond to key blendshapes, from which we extract textures in UV space. Using a subject's expression blendshapes at run-time, we can linearly blend these key textures together to achieve the desired appearance. Furthermore, we can use the mouth interior and eye textures produced by our network to synthesize on-the-fly avatar animations for those regions. Our work produces state-of-the-art quality image and video synthesis, and is the first to our knowledge that is able to generate a dynamically textured avatar with a mouth interior, all from a single image.

CCS Concepts: • **Computing methodologies** → *Computer graphics; Image manipulation*;

Additional Key Words and Phrases: Digital avatar, Texture synthesis, Image-based rendering, Generative adversarial network, Facial animation.

ACM Reference format:
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 258 (November 2018), 12 pages.
https://doi.org/10.1145/3272127.3275075

0730-0301/2018/11-ART258 $15.00
https://doi.org/10.1145/3272127.3275075

1 INTRODUCTION

Recent advances in single-view avatar creation have facilitated the consumer accessibility and scalable production of compelling CG avatars, with potential applications in personalized gaming, social VR, and immersive communication. Such images are easy to access and consumer friendly - all you need is a selfie. In addition, such technology is especially useful for instances in which controlled capture is not available - for example, celebrities or the deceased. Photorealistic CG characters are known to be difficult and expensive to produce, especially in the context of conventional production pipelines. The slightest inaccuracy in the modeling, shading, and

rendering can result in the uncanny valley effect. Image-based approaches, where photographs of subjects are directly used as textures, can easily achieve convincing results while keeping rendering costs low, but are not suitable for reproducing complex non-linear facial expressions or renderings with novel illumination conditions.

Existing methods that can capture facial expression and appearance variations rely on multiple input images which correspond to a set of key expressions [Cao et al. 2014, 2016; Casas et al. 2016; Li et al. 2010; Weise et al. 2011, 2009]. A single input image, however, does not provide dynamic user-specific texture variations. If one were to fit and animate a model using only the texture extracted from this image, wrinkle shadings and other high-frequency appearance variations would not appear. Predicting realistic-looking and plausible skin variations from just one neutral expression requires a comprehensive knowledge of how an entire population would deform their faces.

As observed in Li et al. [2009], static and dynamic details need to be separately treated. Static details are for example moles, nasolabial folds between the cheeks and the upper lip, or periorbital wrinkles, while dynamic details are caused by the shading changes due to skin deformations and blushing which occur during the performance of an expression. Different expression deformations are extremely intricate and appear at different places depending on the subject and hence cannot be interpolated linearly across different individuals. In this work, we also treat the mouth interior and eyes as "dynamic" details, as they share these qualities. Figure 2 shows that synthesized dynamic details are critical for realistic avatar appearance especially for the mouth interior. In order to predict realistic and plausible appearance variations, we need to be able to learn how these wrinkles and appearance variations form for a large number of subjects when performing different expressions.

Inspired by recent advances in generative adversarial networks (GANs) for synthesizing photoreal faces and expressions [Choi et al. 2017; Ding et al. 2017; Karras et al. 2017], we propose a deep learning approach that can synthesize person-specific facial expressions. Given a single face input image in a neutral pose, our system can generate novel photoreal expressions from alternate viewpoints, as well as their corresponding high-resolution UV texture maps which include both the eyes and mouth interior. Our network is also able to produce controllable video sequences directly from its output with a reasonable degree of temporal coherence.

Previous methods [Averbuch-Elor et al. 2017] [Olszewski et al.] that attempt this in the single-image case rely on transferring deformations on-line from a source actor to the target image. However, these can result in transferring inappropriate texture and pigmentation details from the actor, often resulting in creepy-looking facial animations. This is especially true when transferring the mouth interior. In contrast, our method does not transfer but rather synthesizes plausible deformations, including the mouth interior.

While GANs are particularly effective at synthesizing photoreal images, they are also notoriously difficult to control finely. For instance, existing work is able to generate plausible facial expression modifications, but might also result in undesirable appearance attributes, such as adjustments to skin tone and lighting. To deal with these issues, we introduce a photoreal avatar GAN (paGAN). paGAN's network structure allows for fine-grained control by using

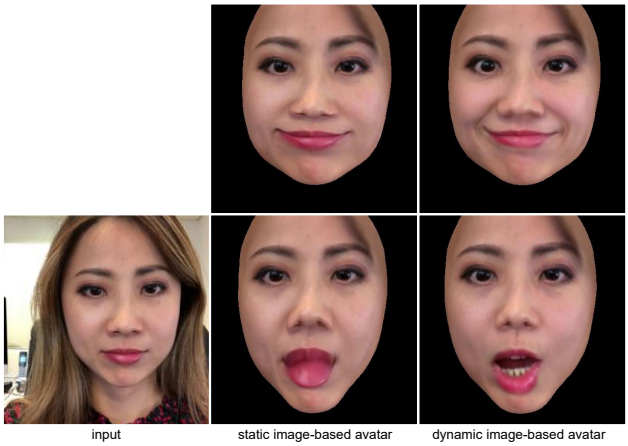

Fig. 2. Comparison with static texture avatar. Static textures do not allow the formation of wrinkles and the mouth interior when the face changes pose.

a combination of rendered expressions, depth/normal maps, and eye-gaze indicators as conditions. These additional controls are computed by fitting a 3D morphable model to the input image. Our network is trained using a massive face image dataset that captures a wide range of subjects and expressions.

Unfortunately, paGAN cannot run in real-time on current mobile devices, due to limited GPU performance. However, once the network is trained, we can use it to generate a sparse set of key expression textures. This sparse set can then be separated into a comprehensive set of blendshape-driven UV texture maps based on the Facial Action Coding System (FACS)[Ekman and Friesen 1978]. These textures are then used to produce compelling image-based avatars with dynamic facial expressions, which can be driven by blendshapes on a mobile device at full 30 fps. We further develop an image-based approach to synthesize photoreal textures of the mouth interior and eye regions of the avatar using paGAN.

Our proposed technique offers several distinct benefits. As shown by Cao et al. [2016], image-based facial blendshapes such as ours do not require complex skin deformations or reflectance separations in the form of diffuse and specular maps, which are needed for sophisticated rendering pipelines. Unlike the previous work of [Cao et al. 2016], we do not require multiple input images, but only a single photo. Furthermore, as we can compute all relevant assets offline, the dynamic-texture online retargeting can be performed in real-time with minimal computational resources in a graphics engine such as Unity or even on a mobile device.

This work is the first to our knowledge that produces dynamically textured avatars, including photorealistic mouth interior and eye regions, from a single image. Our comparisons show that paGAN generates visually superior and higher resolution images than other deep learning alternatives.

Our contributions can be summarized as follows:

- The first image-based dynamic avatar creation framework from a single photograph and end-to-end deep learning approach for the digitization and rendering of photorealistic faces.

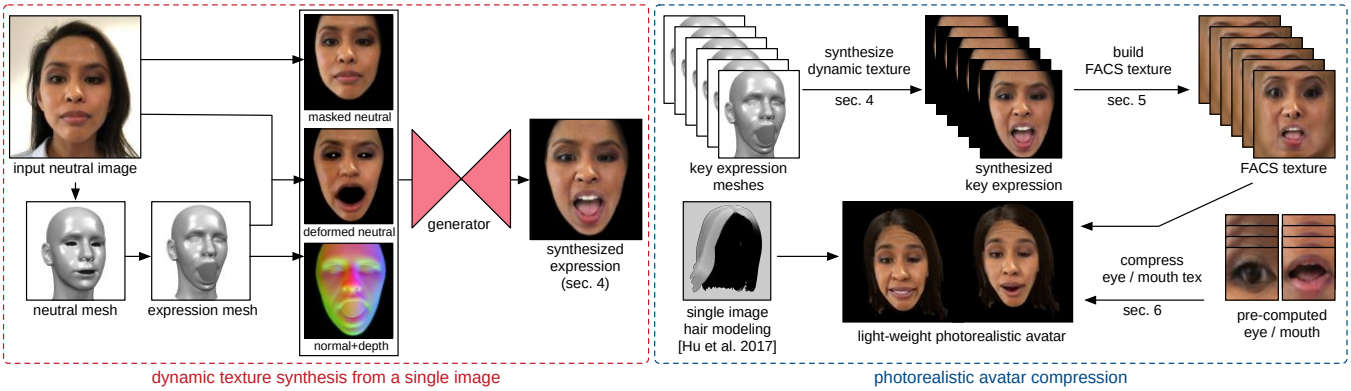

Fig. 3. System Overview. Left: we train GAN that can produce novel expressions in various viewpoints given a neutral face image, and conditioning parameters. Right: extracted textures from our trained network are used to drive a dynamic avatar in real-time on a mobile device.

- We develop a novel deep generative model that can synthesize photorealistic videos of arbitrary facial expressions and views given a single input image of a person. We introduce a novel conditional GAN that is controlled using estimated depth and normal maps, as well as a loss function that preserves the identity of the subject.
- Our synthesis network can generate plausible mouth interior and eye regions from a still portrait, while ensuring photorealistic and person-specific deformations of the subject, whereas existing techniques directly transfer expressions from the source.

2 RELATED WORK

Facial Expression Modelling. The work of [Blanz and Vetter 1999] is the first instance of learning a 3D PCA model of face shape and texture, using around 200 subjects in neutral poses. In order to model the variation in expressions, previous work [Amberg et al. 2008] combine a PCA model of a neutral-expression face shape with a PCA space derived from the residual vectors of different expressions to the neutral pose. More recent techniques [Booth et al. 2017, 2016] are able to learn a linear face model of a neutral expression using around 10,000 scans. Vlasic et al. [2005] use a tensor-based model to jointly represent variations in identity and expression, and Yang et al. [2011] build a separate PCA model per facial expression. Cao et al. [2014] released a comprehensive bi-linear face model database using depth-sensor captured data. Since these methods capture the texture variations by blending linearly between subjects and expressions, they cannot handle high-frequency details such as wrinkles. Nevertheless, these dynamic face models are suitable for single-view face modeling and tracking applications [Blanz and Vetter 1999; Bouaziz et al. 2013; Cao et al. 2014, 2016; Garrido et al. 2016; Hsieh et al. 2015; Hu et al. 2017; Li et al. 2013; Saito et al. 2016, 2017; Thies et al. 2015, 2016a; Weise et al. 2011].

Facial Expression Capture. While only for the geometry of facial expressions, Weise et al. [2009] recorded hundreds of frames of a subject's performance to build its PCA expression model, before tracking. Li et al. [2010] later developed an example-based blendshape modeling technique that only requires a sparse set of key expressions as input to generate a full set of FACS-based expressions. This technique has been adopted in the Kinect-driven facial performance capture framework [Weise et al. 2011] and its commercial implementation Faceshift. Using a dedicated expression scanning session using a high-end 3D acquisition system, previous work [Cao et al. 2015] demonstrates a method that can simultaneously model a face's overall geometry, as well as fine-scale geometry details, such as wrinkles. The authors of [Casas et al. 2016] show that photorealistic looking face models can be built easily using dynamic textures obtained from multiple RGB-D acquisitions. Cao et al. [2016] only use an RGB sensor to create an image-based 3D avatar with dynamic facial textures, but require multiple images of the user and some manual input. In the context of video face replacement, previous work [Dale et al. 2011] use both a re-timed source and target video to warp the facial performance between subjects. To model skin tone variations such as blushing, Jimenez et al. [2010] build a dynamic appearance model of skin from in vivo measurements of melanin and hemoglobin concentrations.

Facial Detail Transfer. Without the need for capturing additional expressions, one way to produce fine-scale texture details during a performance capture is to transfer high-frequency appearance variations from the source video to a target. The method of [Olszewski et al.] is able to transfer expression textures from a performer to a target 3D face for a video-based face replacement application. In particular, this method uses a conditional GAN that operates in the UV texture space. Averbuch-Elor et al. [2017] animate a still portrait using a source actor, but transfer the dynamic details (e.g., wrinkles) from the source, which are not necessarily compatible with the facial features of the portrait. While the above two methods only need a single target image, they copy the mouth interior from the source, which may result in uncanny results.

Facial Texture Synthesis. Instead of transferring details from a source to a target, another way of generating details is to synthesize details from the input image directly. Saito et al. [2017] uses a

pre-trained classification network to synthesize complete and photo-realistic textures by using feature statistics of multiple layers [Gatys et al. 2015].

Isola et al. [2016] are the first to show that one can learn "translation" functions using GANs which map images from one domain to another. Recently, this kind of image-to-image translation framework is applied to capture high-fidelity facial reflectance and geometry [Huynh et al. 2018; Yamaguchi et al. 2018]. Zhu et al. [2017] extend the work of [Isola et al. 2016] by showing that such translation is possible when ground-truth pairing examples are absent. In both these works, however, each domain transfer requires a separate function. Choi et al. [2017] show that, in fact, a single model is able to perform image-to-image translation for multiple domains. However, they are limited to synthesizing a sparse set pre-defined semantic expressions such as smiling, frowning, etc. More recently, Kim et al. [2018] are able to generate controllable performances if provided with a video of the target actor. In contrast, our method can do so using only a single image. Song et al. [2017] use fiducial points, which describe facial geometry, as a conditioning factor as the input to a GAN for outputting varied expression. Their method is confined to a low resolution, however, at 128×128. Furthermore, these landmark points are sparse and therefore fail to capture distinct facial identities and expressions. In contrast, our method uses dense normal and depth maps as conditioning factors which are much more capable of capturing the fine geometric details of a particular expression. The authors of [Ding et al. 2017] show that an expression code module can be learned alongside an encoder-decoder framework, which allows a user to control the intensity of an expression. Karras et al. [2017] recently propose a method of optimizing GANs progressively, adding new layers to both the discriminator and generators throughout the training.

3 METHOD OVERVIEW

Dynamic Texture Synthesis. Given a single face image in a neutral pose, along with a desired blendshape expression and viewpoint, our deep generative network (paGAN) is able to produce a realistic image of the face with the desired expression (Section 4). Our network is conditioned on multiple inputs, including eye gaze and a rendered image of normals of the 3D fitting to the target image, providing fine scale control of the output. We note that we use a single network that is able to work for all identities, whereas the method of [Kim et al. 2018] use different networks trained for each target avatar, requiring for each a source video as training data. Likewise, though Olszewski et al. [Olszewski et al.] can produce animations from a single image of new subjects, they use temporally aligned video performances for training. In either case, a tedious process is required to accommodate new subjects or extend the training set size. Our method works for any number of subjects on a single network, and only requires lightweight annotation of the data (which images have the same identity). Once trained, our network can produce fully-controllable temporally stable video performances from any image. Please see Figure 3 (left half) for details.

Image Based Dynamic Avatar. Once this network is trained, we are able to perform real-time avatar manipulation on a mobile device. We employ a strategy of "compressing" our learned model so that it can run with minimal computational resources, as current mobile devices cannot incorporate the full paGAN on their GPU. To achieve this, we generate a realistic image for each of a set of predetermined expressions, K, that we call "key expressions" (Section 5). Then, using a fitted model to the target, we extract textures in UV space for each of these "key expressions." We use $K = 6$ key expressions for all experiments.

Fig. 4. View-aware texture synthesis evaluation. "View-independent synthesis" shows the result of using a frontally-synthesized texture placed on a fitted mesh turned sideways, whereas view-dependent synthesizes the side-view face directly from paGAN. Distortion in the mouth region is clear for the former approach.

Once these "key expression" textures are computed, we are able to expand them via activation masks into a larger set of FACS-based textures, each corresponding to a specific blendshape vector. We are able to produce 51 textures with the activation masks. Then, by reading the expression blendshape coefficients of a user, we can replace both the geometry and appearance of said user with the likeness of the target by linearly combining the FACS textures.

We note that these key-textures are computed with our network offline and can then be used for real-time applications with a mobile device. We further note that even high-end real-time desktop game characters (e.g., Digital Ira [Alexander et al. 2013]), use only up to eight key expressions, which is further decomposed into FACS-based blendshape textures.

We also use an image-based method for rendering the eyes and mouth interior of the dynamic avatar. We use the trained paGAN to pre-generate multiple textures of both the eyes and the mouth interior, which we then pass to a mobile device (Section 6). For the eye region, we retrieve the nearest eye texture based on the gaze signal from the mobile phone facial tracker to represent dynamic eye gaze. For the mouth, we use a per-pixel weighted median approach [Suwajanakorn et al. 2017] to synthesize appropriate dynamic mouth interiors. Finally, we incorporate the method of [Hu et al. 2017] to add hair to the final model. Please refer to Figure 3 (right half) for details.

4 DYNAMIC TEXTURE SYNTHESIS

Given a frontal faced image I, we first fit a 3D Morphable Model to the image using the method of [Thies et al. 2016b] to obtain the initial mesh $M_I = (\alpha_I, \beta_I, R_I)$, where α_I and β_I are the respective identity and expression coefficients of the fitting to image I, and R_I encodes the orientation parameters (rotation and translation). We then compute the face texture T_I which is unwrapped from I to UV-space. Unlike Thies et al. [2016b], the mesh we use is not hollow behind the eyes and the mouth.

For an arbitrary target expression E with blendshape coefficients β_E and rigid transform parameters R_E, we drive M by replacing the expression blend-shape coefficients to obtain the mesh $M_E =$

(α_I, β_E, R_E). Our dataset contains faces varying up to 45 degrees in every direction from the neutral pose, which allows us to produce photo-realistic facial performances with a large range of motion. We train to output varying viewpoints in order to produce convincing side-view textures for our final avatar. Rendering a straight-on texture from a side-view results in artifacts (see Figure 4).

The input of the network consists of

$$A_I = (I, \Phi(M_E), \delta(M_E), \Gamma(I), \rho(M_E, T_I))$$

, where $\Phi(M_E)$ is an image of the rendered normal directions of M_E, $\delta(M_E)$ is the depth map, $\Gamma(I)$ is a masked-image encoding the the direction of gaze, and $\rho(M_E, T_I)$ is a rendered image of M_E using the input texture T_I, which we call the "deformed neutral" image. We use the camera space of M for each input, so all images are aligned.

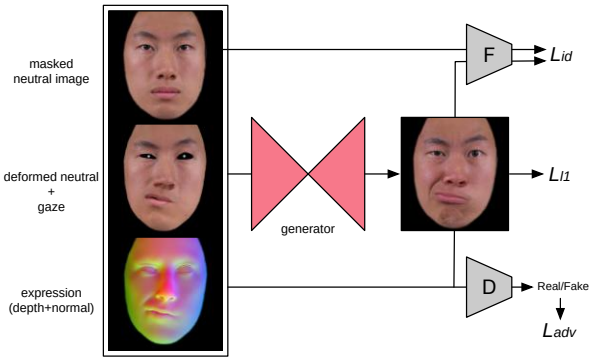

Fig. 5. Illustration of cGAN with discriminator for expression synthesis.

Using this input, we train an image translation network [Isola et al. 2016] to infer the real face image with the correct facial deformations, I_E. The intuition here is that I_E and $\rho(M_E, T_I)$ will be relatively close already. Static texture details should remain aligned, and dynamic deformations caused by the expression β_E can be explained well by the fitted mesh's normals and depth. Furthermore, the normal-image provides a dense pixel-wise orientation map, which is important for the realistic generation of view-dependent eyes and mouth textures. See Figure 5 for architecture illustration and input examples.

More succinctly, we train a U-net based generator G with skip connections [Isola et al. 2016] that tries to generate the true expression image I_E from the input $(I, \Phi(M_E), \delta(M_E), \Gamma(I), \rho(M_E, T_I))$. Our loss is given by:

$$L = \lambda_{adv} L_{adv} + \lambda_{id} L_{id} + \lambda_{\ell_1} L_{\ell_1} \quad (1)$$

, where L_{adv}, L_{id}, and L_{ℓ_1} are the adversarial, identity-preserving, and the pixel-wise Manhattan reconstruction loss, respectively.

Adversarial Loss. We train a patchGAN discriminator D [Isola et al. 2016] that attempts to distinguish between the real hextuple

$$(I, \Phi(M_E), \delta(M_E), \Gamma(I), \rho(M_E, T_I), I_E)$$

, and the generated one

$$(I, \Phi(M_E), \delta(M_E), \Gamma(I), I_{E_{gt}}, G(\cdot))$$

, where $I_{E_{gt}}$ refers to the rendered blendshape fit to the ground-truth image of the actor performing expression E. The adversarial loss is given by $log(D(G(\cdot))$

Pixel-wise Loss. The pixel-wise loss is defined as $||G(\cdot) - I_E||_1$, the sum of pixel-wise absolute differences between the generated expression image and ground-truth expression image.

Identity-Preserving Loss. We use the pre-trained model of Light CNN[Wu et al. 2015] to compute a 256 dimension feature vector encoding the identity of the subject's face, and employ the following identity preserving loss to enforce the identity-likeness between I_0 and the generated image:

$$\mathcal{L}_{id} = \|F(I) - F(G(A_I)\|_1 \quad (2)$$

5 BUILDING FACS TEXTURES

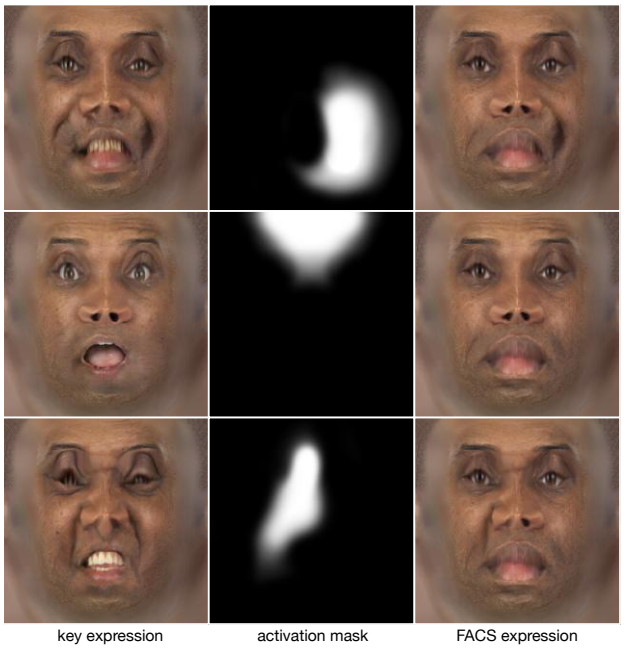

Fig. 6. FACS expressions and corresponding activation masks.

Once the network is trained, we are able to generate any expression we would like for a target neutral image, and can synthesize per-frame textures from any sequence of blendshapes. However, this method is computationally intensive, as we would have to render the condition images (see Figure 5) and run the GAN synthesis at every frame. This needs a high-end desktop GPU and would make real-time applications incredibly difficult on mobile devices. Furthermore, failures at a given frame could result in temporal inconsistencies and undesirable artifacts.

Now, suppose instead that we had a texture map T_e corresponding to each blendshape $e \in \mathcal{E}$, where $\mathcal{E}$ is the set of blendshapes in our model, each of which corresponds to a FACS action. We could then use a linear combination of these textures, weighted by the

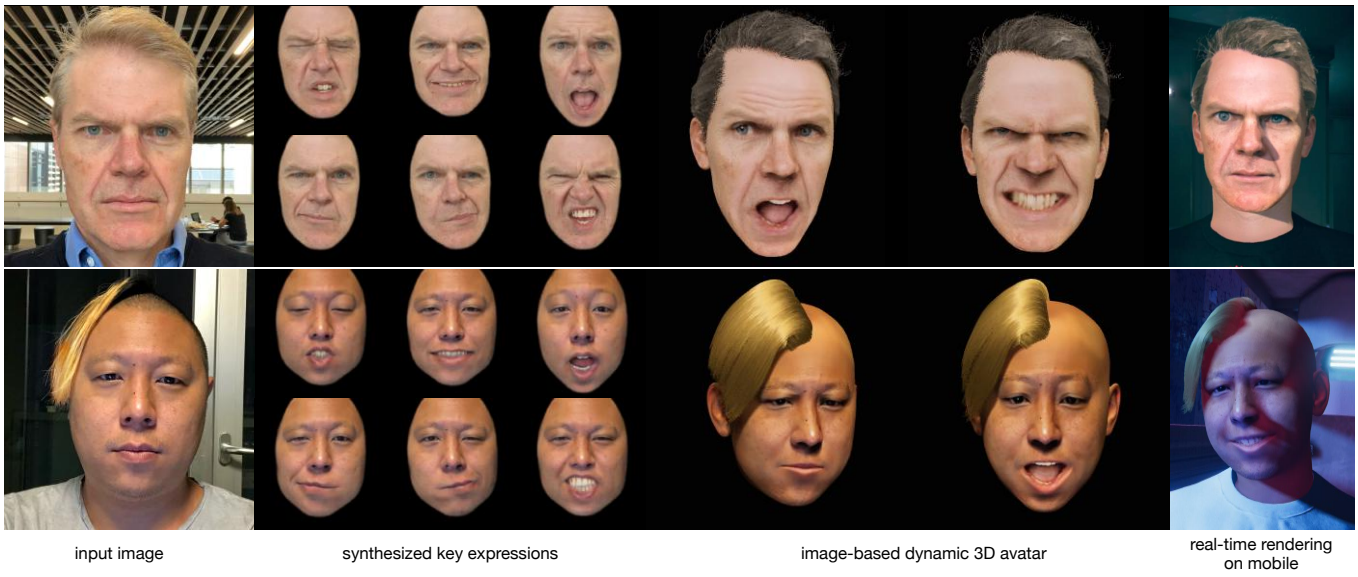

Fig. 7. Gallery of results of image-based dynamic avatars. Avatar hair generated using the method of [Hu et al. 2017]. The far right column shows the avatar rendered for a real-time application in a scene for mobile devices.

expression blendshape coefficients of the fitting, to construct the output texture map we wanted. In this case, a single offline pass could get us the textures we needed for real-time synthesis, without further need of the network. In addition, we could train the network to focus on producing these expressions in particular, decreasing the risk of error. In actuality, the process to achieve this is more complex. First of all, a naive linear blending of entire UV textures won't actually work, as the facial appearance changes locally. Instead, inspired by previous work [Seol et al. 2011], we further apply a UV activation mask per expression which is computed by taking the per-vertex deformation magnitude of each expression from the neutral pose.

That is, given a blendshape mesh $e \in \mathcal{E}$ and a neutral mesh M, its activation mask A_e at vertex v in UV space is defined as follows: $A_e(v) = ||e(v) - M(v)||_2$. That is, the mask's value for vertex v is the magnitude of the deformation at that vertex. The vertex deformation is interpolated to neighboring pixels in UV space so there are no holes or gaps in the activation mask. Also we apply small Gaussian blur on it to remove any discontinuities. A visualization of the activation maps of different expressions can be found in Figure 6.

Then, the final texture at pixel v for expression blendshape coefficients $\{a_e^t\}$ at frame t can be computed as follows:

$$c(v) = w_0(v) \cdot T_0(v) + \sum_{e \in \mathcal{E}} w_e(v) \cdot T_e(v) \tag{3}$$

, where w_e indicates an activation mask modulated by a blendshape coefficient, that is, $w_e = a_e^t \cdot A_e(v)$. Likewise, w_0 is a weight of a neutral expression $w_0 = max(0, 1 - \sum_{e \in \mathcal{E}} w_e(v))$. w_0 and w_e are further normalized to satisfy $w_0 + \sum_{e \in \mathcal{E}} w_e = 1$

While this works in theory, directly obtaining an individual FACS blendshape texture T_e is not practical because performances of isolated FACS (e.g., raising a single eyebrow) is incredibly difficult for actors. Instead, we first infer textures for a set $\mathcal{K}$ of "key expressions" that we found to be easily performed and reconstruct the FACS based textures from these. For a key expression $k \in \mathcal{K}$ and corresponding blendshape coefficients $\{a_e^k\}$, an activation weight w_k is given by $w_k = \sum_{e \in \mathcal{E}} a_e^k \cdot A_e(v)$. Now using the "key expressions", we compute the dynamic texture as:

$$c(v) = w_0(v) \cdot T_0(v) + \sum_{k \in \mathcal{K}} w_k(v) \cdot T_k(v) \tag{4}$$

, where w_k is further modulated by time-varying blendshape coefficients $\{a_e^t\}$ as $w_k = \sum_{e \in \mathcal{E}} a_e^k \cdot a_e^t \cdot A_e(v)$ and $w_0 = max(0, 1 - \sum_{k \in \mathcal{K}} w_k(v))$ followed by normalization. Finally, a FACS texture for expression e is recovered by setting $w_k = a_e^k \cdot A_e(v)$ (i.e., $a_i^t = 1$ for $i = e$ and $a_i^t = 0$ otherwise).

6 REAL-TIME IMAGE BASED AVATARS

While Cao et al. [2016] introduce a method to generate dynamic per-frame-textured avatars, they require multiple input images by the user. On the other hand, Hu et al. [2017] demonstrates the ability to reconstruct an avatar from a single image, but is unable to compute dynamic textures for it. Our pipeline, in contrast, is able to both generate a dynamically textured avatar while still only requiring a single image. Furthermore, the avatar can be controlled in real-time from the user's facial performance on a mobile device (see Figure 7).

FACS-based Textures. We pre-compute K (= 6) key expression textures (see previous section) and pass them onto the mobile device afterwards. For efficiency, we implemented Equation 4 on a pixel shader and directly compute the dynamic texture without explicitly storing individual FACS textures. For each frame of the avatar's animation, using the tracked expression coefficients from the user,

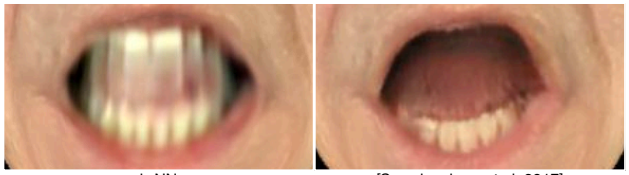

Fig. 8. Alternative mouth synthesis approach using k-nearest-neighbor retrieval and blending. Using the weighted-median approach of [Suwajanakorn et al. 2017] produces sharp teeth while avoiding artifacts.

we are able to synthesize per-frame textures online (as described in Section 5) and apply it to the deformed mesh.

Gaze Control. The eyes are dealt with in a similar manner: we pre-compute 20 eye-textures for the avatar that approximate all viewing directions using our paGAN and pass them to the mobile device. Using the gaze tracker on the mobile device, we pick an eye texture with the nearest gaze direction and composite it onto the UV face texture.

Mouth Interior Synthesis. First, we pre-compute L (up to 300) mouth interior textures that correspond to a large variation of mouth poses using the dynamic texture synthesis in Section 4. Next, instead of using a naive nearest-neighbor retrieval, we employ the per-pixel weighted median blending of [Suwajanakorn et al. 2017]. Out of L pre-computed textures, 50 nearest neighbor (NN) frames are used for the weighted median blending, which is implemented on a pixel shader. Following the implementation of [Suwajanakorn et al. 2017], the NN frames are determined and weighted based on the correlation of the sparse mouth landmarks between the current and the L database frames. The database frames are chosen from a subset of synthesized talking and range of mouth poses so they cover sufficient deformation and appearance variations. A comparison of different mouth synthesis methods for a compressed avatar can be found in Figure 8.

7 DATA COLLECTION AND TRAINING DETAILS

We use a combination of datasets for training and evaluating our system. We list them below:

- Our proprietary data consisting of 67 subjects with differing numbers of expressions and viewing angles, up to 40 per individual.
- 158 out of 597 available subjects from the Chicago Face Dataset (CFD). We choose subjects that have multiple expressions within the dataset [Ma et al. 2015] and use the rest for testing.
- 230 subjects with 22 expressions each from the compound facial expressions (CFE) dataset [Du et al. 2014].
- 14 subjects with 30 expressions each from the work of [Olszewski et al.].
- 23 subjects with 15 expression each from the ICT 3DRFE dataset [Stratou et al. 2011].
- 67 subjects, each with 3 gazes and 3 camera angles per each of 8 expressions, from the Radbound Faces dataset [Langner et al. 2010].

Table 1. Photometric error from ground truth. Best performance is achieved with full model.

input	MSE	SSIM	PSNR
depth/normal	299.5	0.9406	24.74
deformed neutral	260.1	0.9380	25.37
combined	**102.5**	**0.9783**	**28.74**

For training data, our network only needs a set of unconstrained expression photographs and does not require a complex capture setup. Our data is automatically annotated using a face fitting process described in Section 4. After the initial face fitting, we mask out the non-face region of the input using the fitted 3D geometry (see Figure 3 left). We additionally apply some feathering to remove the hard edges on the mask. These pre-processed images are then re-sized, rotated, and translated so the faces are aligned on top of each other. We train on 9,000 images and leave 1,000 aside for testing. We train for 200 epochs on an NVIDIA Tesla V100 GPU using the Adam optimizer [Kingma and Ba 2014] with an initial learning rate of 0.0001 in PyTorch and a batch-size of 32. We weight our loss function using $\lambda_{adv} = 1$, $\lambda_{id} = 0.15$, and $\lambda_{\ell_1} = 10$. Our network outputs at a resolution of 512×512 pixels.

We found that high quality dense annotations using the deformed neutral, normal and depth maps obtained from the 3D face fitting [Thies et al. 2016b] (see Figure 5) were necessary for stable training and high-quality temporal coherent output from the GAN. Using these dense conditions, the network is able to focus on learning expression-specific details, such as wrinkles, mouth, and eye textures, based on the shape condition. Hence, we can avoid ambiguity in the training, which could otherwise lead to corrupted results. We also note that the training result depends on a balanced set of data and ensure that the number of training photos is similar per subject and the variations of the expressions are well distributed.

8 RESULTS

Figures 1 and 7 show synthesized key expressions and instances of our dynamic avatars, after FACS textures have been computed (outlined in Section 5). Our method is able to reproduce sharp wrinkles and creases in the avatar's complexion when being driven by blendshapes. Refer to the supplementary video for further examples. A gallery of network results can be found in Figure 9. Notice that our network is able to synthesize highly variable expression details including wrinkles and the interior of the mouth at different viewing angles.

Comparisons. Figure 10 shows comparisons with ground truth expressions. A visualization of a per-pixel Euclidean distance error on example instances can be found in Figure 11. We also provide a table for the photometric error in Table 1. It is easy to see that our method produces much perceptually superior results than other state of the art GAN approaches such as Song et al. [2017] and Choi et al. [2017] which have noticeable artifacts (see Figure 12).

The difference in quality compared with Song et al. [2017] is especially noteworthy, given that both our method and theirs condition

Fig. 9. Gallery of synthesized expressions. Note that only the blendshape fitting of the source is used to produce these results - no transfer of wrinkles/mouth from the source image is used. Original image courtesy of Getty Images (target 3 and 4).

GANs on geometric properties. We thus posit that using dense geometric indicators, such as the pixel-wise normals and depth, gives much more control when working with GANs than sparse fiducial landmarks. This hypothesis is reinforced when comparing with a version of our method that omits using either the deformed neutral or the normal/depth maps as an input (see Figure 13). When omitted, fine scale features such as the mouth interior do not appear at all.

A comparison with expression transfer methods can be found in Figure 14. These can sometimes produce results found in the source image that are not appropriate in the target. We show other avatar methods in Figure 15 and Figure 16. The avatars generated in 15 require multiple input images while the method of [Hu et al. 2017] in Figure 16 does not have dynamic textures.

Performance. Table 2 lists the running time for each stage of our pipeline. Our offline experiments are performed using an Intel Core i7-5930K CPU with an NVIDIA Tesla V100 GPU. Our real-time experiments are performed on an Apple iPhone X. Note that while the trained network takes only 7 ms on the desktop for texture synthesis, directly porting such a network onto a mobile is impractical. Our proposed method, on the other hand, allows real-time performance on mobile devices.

User Study. We conduct an Amazon Mechanical Turk study asking participants to distinguish real face images from the synthetic ones generated by our method. The user study is performed on dynamic expression photos since the main focus of our GAN is to provide photorealistic key expressions for our real-time avatar. The questionnaire is posed in the following format: each question

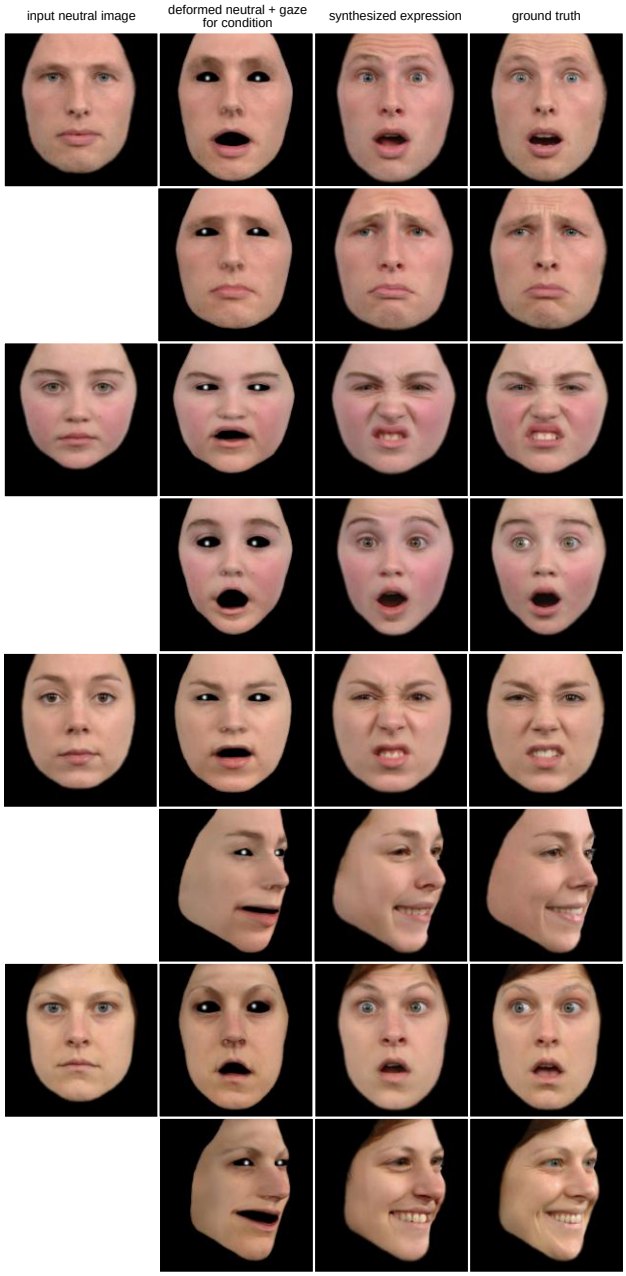

Fig. 10. Gallery of synthesized results compared with ground truth. Though synthesized deformations do not always correspond with ground-truth dynamic details, the network-generated expression are remain plausible and realistic due to the adversarial loss. Only the deformed neutral and gaze image is shown for the condition.

contains 5 face images, 4 of which are real and 1 of which is generated by our network. Examples of real and synthesized images are given before the questionnaire, and the tester is asked to pick the generated one out of 5 images each time, without a time limit. There are two modes of question, one in which each image is the same identity making different expressions, and one in which each image is of a different individual making the same expression.

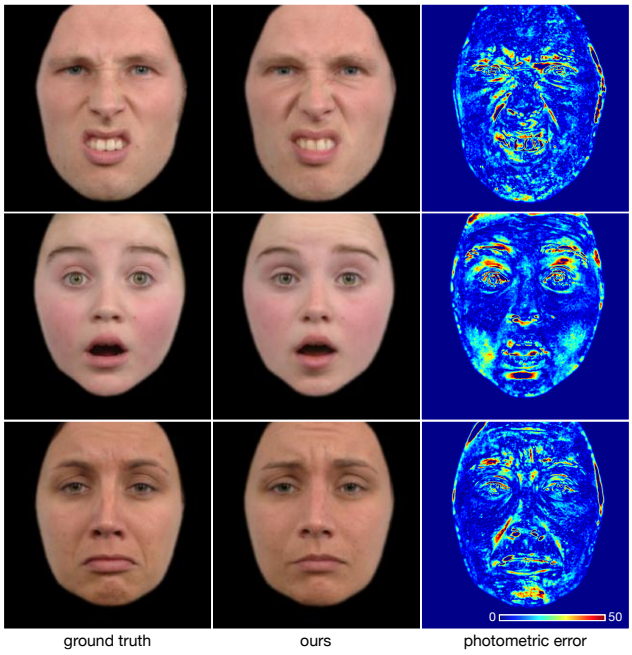

Fig. 11. Visualization of a photometric error compared to ground truth.

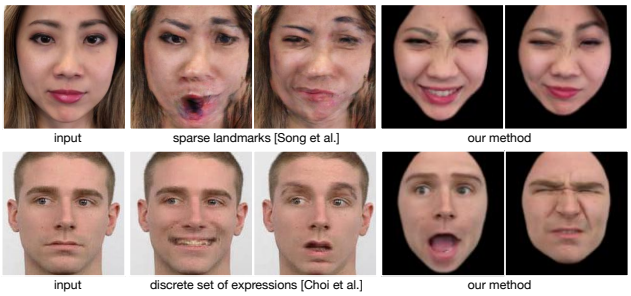

Fig. 12. Comparison with other GAN-based texture synthesis methods. Other methods produce noticeable artifacts.

If a user is not able to tell the difference, s/he should be getting the correct answer at around a **20.0**% rate with random guessing. We asked each of 200 users 3 of these questions, constructed at random. The average rate of choosing correctly was **26.6**% ± **0.054**% for same expression type questions and **25.7**% ± **0.036**% for same identity type questions and nobody was able to identify all the synthesized images. See Table 3 for details. This indicates that our method nearly completely fooled the users.

9 CONCLUSION

We have demonstrated paGAN - an end-to-end deep learning approach for facial expression texture synthesis - on a wide range of challenging examples. We show how individualized and continuous

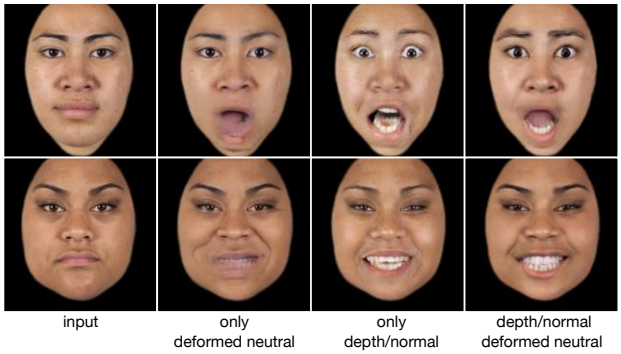

Fig. 13. Qualitative ablation evaluation of paGAN. The second column from left is generated from a network trained using only a deformed neutral face in addition to the input image. The third column from left uses all inputs except the deformed neutral, and the right-most column is the result from the full model.

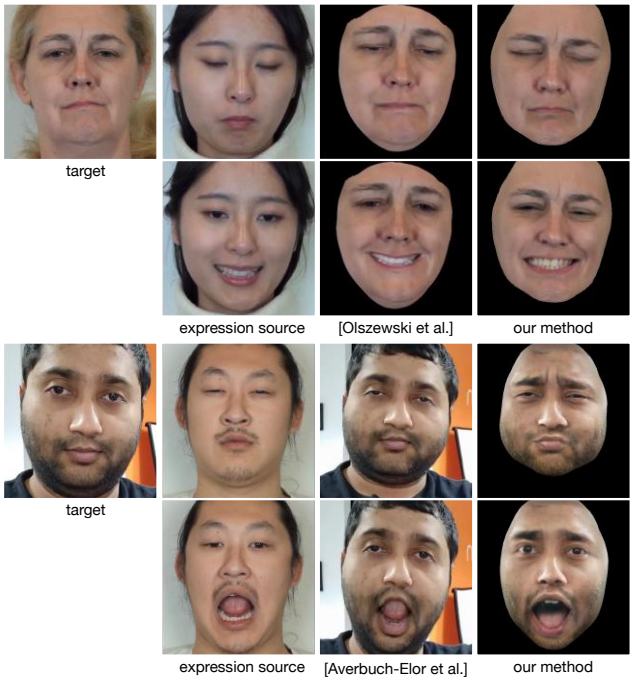

Fig. 14. Comparison with expression transfer methods. Transfer from source can produce inappropriate results. The method of [Averbuch-Elor et al. 2017] (bottom) seems to transfer the source's tongue too high in the target mouth (third column), whereas our method does not use the source tongue.

variations of facial expressions, including the mouth interior and eyes, can be synthesized in various poses from a single input image. Our technique does not train individualized networks for each identity [Kim et al. 2018], meaning new subjects are easy to process. We show that paGAN enables performance-based animation using an image-based dynamic avatar as well as video-driven facial animation, by generating a compressed model representation that can be run on a mobile device in real-time.

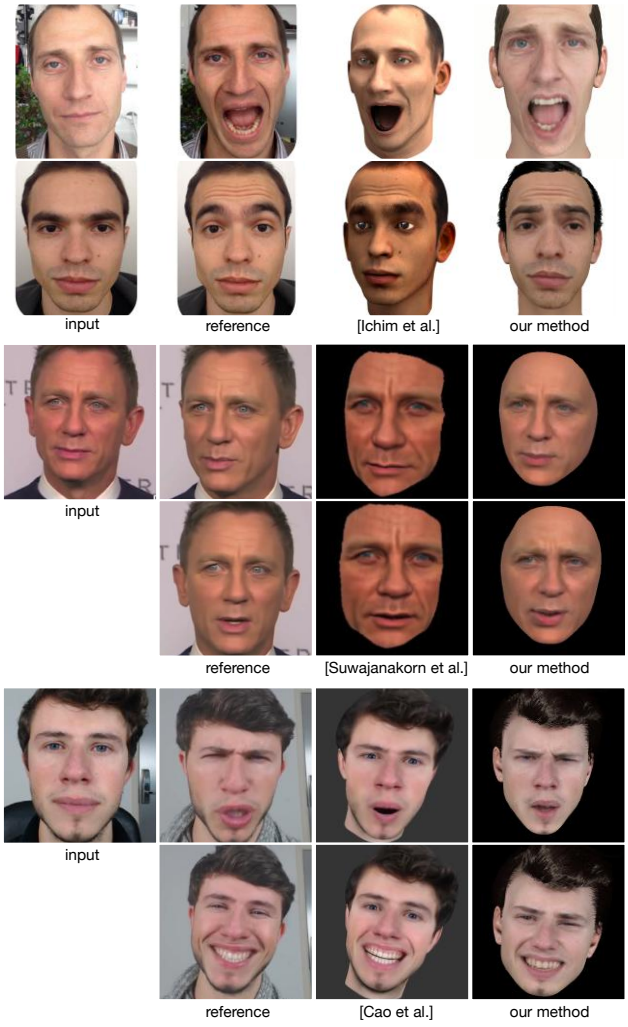

Fig. 15. Comparison with other avatar methods. Other methods require multiple input images, whereas we only require one. The second method comparison is rendered without hair and ears for more direct comparison with Suwajanakorn et al. [2015]. Daniel Craig images courtesy of Getty Images.

Limitations. As mentioned earlier, our method requires the input face to be roughly frontal, in a neutral pose, and well illuminated. Our method mostly fails, when the subject has a three quarter view face, is smiling, or when a hard shadow is cast on its face (Figure 17 first and second columns). Furthermore, we cannot handle a tongue sticking out and occlusions from hair, glasses, or hands (Figure 17 third and fourth columns). Also it is not possible to predict the exact person-specific dynamic appearance (e.g., same number of teeth) from just a single still portrait (Figure 18).

Future Work. There are many future avenues to extend our work. Video re-enactment [Kim et al. 2018; Thies et al. 2016b] has been shown to be possible given a video as input. These re-enactments

Fig. 16. Comparison with Hu et al. [2017]. They do not used an image-based approach for eyes and teeth, but generic 3D models instead, resulting in an uncanny appearance.

Table 2. Running time of each stage. Creating textures per frame is very slow on a cell-phone, while our modified pipeline runs at 30 fps on an iPhone X. For hair modeling, we are using the code from [Hu et al. 2017] without optimization.

stage	time (ms)
neutral face fitting	500
hair modeling [Hu et al. 2017]	100,000
texture synthesis (desktop)	7
texture synthesis (mobile)	350
tracking (mobile)	10
texture composition (mobile)	4
mouth interior synthesis (mobile)	15

Table 3. Fooling rate of user study. Our method performs very close to being able to fool the users absolutely.

Task	Fooling rate
random guessing	0.800
different expressions	0.743±0.036
different individuals	0.734±0.054
perfect distinguisher	0.000

include the background and body of the target avatar. The natural next step is to do this from a single image. It would also be interesting to collect data with accurate lighting annotations; using lighting as a condition would be especially useful for the mobile application, allowing us to relight the face to match the user's current environment. Our method works for generic expressions transferred to a user-specific identity, even though the resulting expression blendshape may not accurately predict the ones from this person. In the future, we could jointly predict user-specific shapes in addition to expression textures for more accurate results.

Fig. 17. Failure cases. An input with an angled face and a non-neutral expression can lead to artifacts (first column). The input has a strong shadow, which cannot be removed, when synthesizing a new viewpoint (second column). Occluding objects such as the tongue (third column), the hand, and glasses (fourth column) also cannot be removed in the synthesis.

Fig. 18. The ground truth mouth interior is missing a tooth, but network synthesizes a full suite of teeth.

ACKNOWLEDGMENTS

We would like to thank Mike Seymour, Charlie Warzel, Azia Celestino, Erik Castellanos, Kyle San, Jiale Kuang, Aaron Hong, and Frances Chen for being our subjects, Stuti Rastogi and Han-Wei Kung for helping with generating results, Kyle Olszewski for proofreading and editing, and Averbunch-Elor et al. for the comparisons.

REFERENCES

P. Ekman and W. Friesen. 1978. *Facial Action Coding System: A Technique for the Measurement of Facial Movement.* Consulting Psychologists Press, Palo Alto.

O. Alexander, G. Fyffe, J. Busch, X. Yu, R. Ichikari, A. Jones, P. Debevec, J. Jimenez, E. Danvoye, B. Antionazzi, M. Eheler, Z. Kysela, and J. von der Pahlen. 2013. Digital Ira: Creating a Real-time Photoreal Digital Actor. In *ACM SIGGRAPH 2013 Posters (SIGGRAPH '13).* ACM, New York, NY, USA, Article 1, 1 pages.

B. Amberg, R. Knothe, and T. Vetter. 2008. Expression Invariant 3D Face Recognition with a Morphable Model. In *International Conference on Automatic Face Gesture Recognition.* 1–6.

H. Averbuch-Elor, D. Cohen-Or, J. Kopf, and M. F. Cohen. 2017. Bringing Portraits to Life. *ACM Trans. Graph.* 36, 4 (2017), to appear.

V. Blanz and T. 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).* 187–194.

J. Booth, A. Roussos, A. Ponniah, D. Dunaway, and S. Zafeiriou. 2017. Large Scale 3D Morphable Models. *International Journal of Computer Vision* (2017), 1–22.

J. Booth, A. Roussos, S. Zafeiriou, A. Ponniahy, and D. Dunaway. 2016. A 3D Morphable Model Learnt from 10,000 Faces. In *Conference on Computer Vision and Pattern Recognition.* 5543–5552.

S. Bouaziz, Y. Wang, and M. Pauly. 2013. Online Modeling for Realtime Facial Animation. *ACM Trans. Graph.* 32, 4, Article 40 (July 2013), 10 pages.
C. Cao, D. Bradley, K. Zhou, and T. Beeler. 2015. Real-time high-fidelity facial performance capture. *ACM Trans. Graph.* 34, 4 (2015), 46.
C. Cao, Y. Weng, S. Zhou, Y. Tong, and K. Zhou. 2014. Facewarehouse: A 3d facial expression database for visual computing. *IEEE TVCG* 20, 3 (2014), 413–425.
C. Cao, H. Wu, Y. Weng, T. Shao, and K. Zhou. 2016. Real-time facial animation with image-based dynamic avatars. *ACM Trans. Graph.* 35, 4 (2016), 126.
D. Casas, A. Feng, O. Alexander, G. Fyffe, P. Debevec, R. Ichikari, H. Li, K. Olszewski, E. Suma, and A. Shapiro. 2016. Rapid Photorealistic Blendshape Modeling from RGB-D Sensors. In *Proceedings of the 29th International Conference on Computer Animation and Social Agents*. ACM, 121–129.
Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo. 2017. StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation. *arXiv preprint arXiv:1711.09020* (2017).
K. Dale, K. Sunkavalli, M. K. Johnson, D. Vlasic, W. Matusik, and H. Pfister. 2011. Video Face Replacement. *ACM Trans. Graph.* 30, 6, Article 130 (Dec. 2011), 10 pages.
H. Ding, K. Sricharan, and R. Chellappa. 2017. Exprgan: Facial expression editing with controllable expression intensity. *arXiv preprint arXiv:1709.03842* (2017).
S. Du, Y. Tao, and A. M. Martinez. 2014. Compound facial expressions of emotion. *Proceedings of the National Academy of Sciences* 111, 15 (2014), E1454–E1462.
P. Garrido, M. Zollhöfer, D. Casas, L. Valgaerts, K. Varanasi, P. Pérez, and C. Theobalt. 2016. Reconstruction of personalized 3D face rigs from monocular video. *ACM Trans. Graph.* 35, 3 (2016), 28.
L. A. Gatys, A. S. Ecker, and M. Bethge. 2015. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576* (2015).
P.-L. Hsieh, C. Ma, J. Yu, and H. Li. 2015. Unconstrained realtime facial performance capture. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1675–1683.
L. Hu, S. Saito, L. Wei, K. Nagano, J. Seo, J. Fursund, I. Sadeghi, C. Sun, Y.-C. Chen, and H. Li. 2017. Avatar Digitization From a Single Image For Real-Time Rendering. *ACM Trans. Graph.* 36, 6 (2017).
L. Huynh, W. Chen, S. Saito, J. Xing, K. Nagano, A. Jones, P. Debevec, and H. Li. 2018. Mesoscopic Facial Geometry Inference Using Deep Neural Networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. 2016. Image-to-image translation with conditional adversarial networks. *arXiv preprint arXiv:1611.07004* (2016).
J. Jimenez, T. Scully, N. Barbosa, C. Donner, X. Alvarez, T. Viera, P. Matts, V. Orvalho, D. Gutierrez, and T. Weyrich. 2010. A Practical Appearance Model for Dynamic Facial Color. 29, 5 (2010), 141:1–141:9.
T. Karras, T. Aila, S. Laine, and J. Lehtinen. 2017. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196* (2017).
H. Kim, P. Carrido, A. Tewari, W. Xu, J. Thies, M. Niessner, P. Pérez, C. Richardt, M. Zollhöfer, and C. Theobalt. 2018. Deep Video Portraits. *ACM Trans. Graph.* 37, 4, Article 163 (July 2018), 14 pages.
D. Kingma and J. Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
O. Langner, R. Dotsch, G. Bijlstra, D. H. Wigboldus, S. T. Hawk, and A. Van Knippenberg. 2010. Presentation and validation of the Radboud Faces Database. *Cognition and emotion* 24, 8 (2010), 1377–1388.
H. Li, B. Adams, L. J. Guibas, and M. Pauly. 2009. Robust Single-View Geometry And Motion Reconstruction. *ACM Trans. Graph.* 28, 5 (2009).
H. Li, T. Weise, and M. Pauly. 2010. Example-Based Facial Rigging. *ACM Trans. Graph.* 29, 3 (July 2010).
H. Li, J. Yu, Y. Ye, and C. Bregler. 2013. Realtime Facial Animation with On-the-fly Correctives. *ACM Trans. Graph.* 32, 4 (July 2013).
D. S. Ma, J. Correll, and B. Wittenbrink. 2015. The Chicago face database: A free stimulus set of faces and norming data. *Behavior research methods* 47, 4 (2015), 1122–1135.
K. Olszewski, Z. Li, C. Yang, Y. Zhou, R. Yu, Z. Huang, S. Xiang, S. Saito, P. Kohli, and H. Li. Realistic dynamic facial textures from a single image using gans.
S. Saito, T. Li, and H. Li. 2016. Real-Time Facial Segmentation and Performance Capture from RGB Input. In *ECCV*.
S. Saito, L. Wei, L. Hu, K. Nagano, and H. Li. 2017. Photorealistic Facial Texture Inference Using Deep Neural Networks. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*.
Y. Seol, J. Seo, P. H. Kim, J. Lewis, and J. Noh. 2011. Artist friendly facial animation retargeting. In *ACM Trans. Graph.*, Vol. 30. ACM, 162.
L. Song, Z. Lu, R. He, Z. Sun, and T. Tan. 2017. Geometry Guided Adversarial Facial Expression Synthesis. *arXiv preprint arXiv:1712.03474* (2017).
G. Stratou, A. Ghosh, P. Debevec, and L.-P. Morency. 2011. Effect of illumination on automatic expression recognition: a novel 3D relightable facial database. In *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*. IEEE, 611–618.
S. Suwajanakorn, S. M. Seitz, and I. Kemelmacher-Shlizerman. 2015. What Makes Tom Hanks Look Like Tom Hanks. In *Proceedings of the IEEE International Conference on Computer Vision*. 3952–3960.
S. Suwajanakorn, S. M. Seitz, and I. Kemelmacher-Shlizerman. 2017. Synthesizing obama: learning lip sync from audio. *ACM Trans. Graph.* 36, 4 (2017), 95.
J. Thies, M. Zollhöfer, M. Nießner, L. Valgaerts, M. Stamminger, and C. Theobalt. 2015. Real-time Expression Transfer for Facial Reenactment. *ACM Trans. Graph.* 34, 6 (2015).
J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner. 2016a. Face2Face: Real-time Face Capture and Reenactment of RGB Videos. In *IEEE CVPR*.
J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Nießner. 2016b. Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2387–2395.
D. Vlasic, M. Brand, H. Pfister, and J. Popović. 2005. Face transfer with multilinear models. *ACM Trans. Graph.* 24, 3 (2005), 426–433.
T. Weise, S. Bouaziz, H. Li, and M. Pauly. 2011. Realtime Performance-Based Facial Animation. *ACM Trans. Graph.* 30, 4 (July 2011).
T. Weise, H. Li, L. V. Gool, and M. Pauly. 2009. Face/Off: Live Facial Puppetry. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer animation (Proc. SCA'09)*. Eurographics Association, ETH Zurich.
X. Wu, R. He, Z. Sun, and T. Tan. 2015. A light CNN for deep face representation with noisy labels. *arXiv preprint arXiv:1511.02683* (2015).
S. Yamaguchi, S. Saito, K. Nagano, Y. Zhao, W. Chen, K. Olszewski, S. Morishima, and H. Li. 2018. High-fidelity Facial Reflectance and Geometry Inference from an Unconstrained Image. *ACM Trans. Graph.* 37, 4, Article 162 (July 2018), 14 pages.
F. Yang, J. Wang, E. Shechtman, L. Bourdev, and D. Metaxas. 2011. Expression flow for 3D-aware face component transfer. *ACM Trans. Graph.* 30, 4 (2011), 60:1–10.
J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint arXiv:1703.10593* (2017).