Example-Based Facial Animation of Virtual Reality Avatars Using Auto-Regressive Neural Networks

Wolfgang Paier, Anna Hilsmann, and Peter Eisert, *Fraunhofer-Institut für Nachrichtentechnik Heinrich-Hertz-Institut HHI, 10587 Berlin, Germany*

This article presents a hybrid animation approach that combines example-based and neural animation methods to create a simple, yet powerful animation regime for human faces. Example-based methods usually employ a database of prerecorded sequences that are concatenated or looped in order to synthesize novel animations. In contrast to this traditional example-based approach, we introduce a light-weight auto-regressive network to transform our animation-database into a parametric model. During training, our network learns the dynamics of facial expressions, which enables the replay of annotated sequences from our animation database as well as their seamless concatenation in new order. This representation is especially useful for the synthesis of visual speech, where coarticulation creates interdependencies between adjacent visemes, which affects their appearance. Instead of creating an exhaustive database that contains all viseme variants, we use our animation-network to predict the correct appearance. This allows realistic synthesis of novel facial animation sequences like visual-speech but also general facial expressions in an example-based manner.

Animation and rendering of human faces play an important role in many application fields like virtual reality (VR), video-games, or movie-productions. Especially VR requires high-quality animation of 3-D human faces to ensure a high level of realism. While recent approaches for modeling and animating human faces show impressive results, automatic animation is still a challenging task. This is mainly caused by the fact that the human face is a complex structure composed of different materials like bones, muscles, tissue layers, and skin, which results in complex reflective properties (that even include subsurface scattering) but also complex dynamics and motion. At the same time, humans are very sensitive to deviations from expected appearance/behavior of faces such that even small errors may reduce the acceptance of users.

Therefore, we follow a data-driven approach for facial animation, which focuses on the synthesis of realistic faces based on real measurements and animation from a sequence of discrete labels. These labels are a high-level description of the desired facial animation and represent, for example, visemes or general expressions (e.g., neutral, smile, open mouth). The advantage of our approach is that it requires only a minimal description of the target content, which allows even untrained users or other software modules to produce plausible facial animations in real-time. A typical use-case for our system is, for example, the synthesis of visual speech (mouthings) during sign-language production with a virtual human signer. Visual speech synthesis refers to the process of generating plausible mouth animations according to an input speech signal (e.g., text/visemes). Mouthings and facial expressions in general are an integral part of sign-languages as they help comprehending ambiguous signs or gestures, they provide functions like indexing/spatial referencing (e.g., via eye gaze) and convey emotions. In contrast to other animation methods (e.g., the paper by Zhou *et al.*[1]),

Digital Object Identifier 10.1109/MCG.2021.3068035
Date of publication 23 March 2021; date of current version 12 July 2021.

the main challenge of this approach is to generate realistic facial animations (appearance as well as dynamics) from a few semantic labels that do not provide detailed information of the concrete facial expressions or the dynamics.

We solve this task by employing two neural networks. First, we create a neural face model that is capable of synthesizing realistic facial expressions from a low-dimensional latent expression vector. This face modeling network is implemented as a variational auto-encoder (VAE) and jointly reconstructs 3-D face geometry as well as texture, see Figure 1.

The second neural network is responsible for synthesizing a sequence of latent expression vectors given the current and next animation label as well as a style vector. Intuitively, our approach transforms the animation-database into a parametric model. Instead of querying suitable animation samples from the database, we synthesize them using the animation network. This approach is especially advantageous for the synthesis of visual speech, where coarticulation (a phenomenon in human speech where successive sounds are articulated together) strongly affects the appearance of visemes. Instead of creating an exhaustive database that contains all possible instances of each viseme, our network learns to predict the correct appearance from semantic animation labels. This reduces the number of necessary prerecorded sequences and enables us to synthesize realistic visual speech as well as general facial animation sequences from a rather small animation database. Moreover, we train the network to learn style information for each training sequence based on a unique ID. This style vector captures natural variations of facial expressions with the same label. Figure 8 illustrates this effect: our training data contains different instances of the smile expression and instead of creating a new semantic label for each instance, we encourage our animation network to learn continuous style vectors that capture natural variations of facial expressions with the same semantic label. The main contributions of the proposed system are as follows:

- automatic generation of realistic face animations from high-level semantic labels;
- pronunciation of new words, which are not part of the training data;
- capture and reproduction of effects like coarticulation and natural variations of facial expressions with the same semantic label.

The remainder of this document is structured as follows: after presenting important related works, we will give a high-level overview of the proposed system followed by a detailed presentation of the facial modeling and the animation approach. In the last two sections, we will present animation samples and discuss the results as well as the limitations.

RELATED WORK

Modeling and animation of human faces have a long history in computer graphics and computer vision. For

FIGURE 1. Results of the facial performance capture process. Each row shows a different facial expression. From left to right, we show the dynamic face texture, the underlying face geometry, the rendered face model, and the original video frame.

a long time, dynamic face models were typically built by computing a linear basis that allowed reconstructing 3-D face geometry depending on facial expression and identity parameters. More sophisticated models, as, for example, proposed by Thies *et al.*,[2] were even able to capture albedo and diffuse light by incorporating more model parameters. However, while these models are computationally efficient and easy to use, employing only linear relations for a complex object such as the human face leads to well-known problems like the fact that an unregularized linear model can easily generate implausible facial expressions, while at the same time it lacks the expressiveness to capture important details. One approach to circumvent these limitations is using hybrid models like, for example, presented in the paper by Paier *et al.*[3] By incorporating high-resolution dynamic textures, it is possible to capture and reproduce important facial details that cannot be efficiently represented by geometry alone. While this approach improves the visual quality of the rendered model, it also increases the memory requirements and limits its editing capabilities.

With the advent of deep neural networks, more powerful generative models have been developed. For example, Lombardi *et al.*[4] and Paier *et al.*[5] propose an architecture based on a VAE[6] that learns a manifold of facial expression and is capable of jointly reconstructing face geometry and texture with a deep neural network. While Lombardi *et al.*[4] use a highly specialized face capture rig to generate detailed face geometry and view dependent textures from 40 different camera views, Paier *et al.*[5] propose a practical approach with less hardware requirements based on an approximate PCA geometry model. Potential inaccuracies in geometry are compensated with a graph-cut-based multiview texture extraction method that extracts highly detailed dynamic textures. An additional adversarial loss forces the VAE to generate sharp and detailed face textures. While the two previous approaches are restricted to a single person, Li *et al.*[7] and Chandran *et al.*[8] propose deep neural face models, which are capable of representing facial expression and identity. Chandran *et al.*[8] train their neural face model with registered and textured face meshes of 224 subjects that show 24 predefined expressions. They use two separate VAE to extract identity and expression information. The identity VAE processes the neutral expression of each subject, while the expression VAE receives only blend-shape weights. A joined decoder uses the latent identity and expression vector to reconstruct the target geometry as well as a low-resolution albedo map. Using a super resolution network, they compute the final high-resolution albedo map. Li *et al.*[7] build their neural face model using two generative adversarial networks (GAN) based on the styleGAN architecture. An identity GAN generates low-resolution (256×256) albedo and geometry maps in texture space. The synthesized albedo and geometry maps are evaluated by three discriminators. One discriminator checks the albedo map, another one evaluates the geometry map, and a joint discriminator ensures that the generator learns the correct correlation between geometry and albedo. The second GAN generates an expression offset map from a latent expression vector. Again, a discriminator evaluates the synthesized expression map. Additionally, they use an expression code regressor that predicts the latent expression code from the synthesized expression map and minimizes the L1 loss between predicted and true expression code. With two more upscaling steps, they finally generate 4 K albedo, specular and displacement maps from the synthesized low-resolution albedo and expression-geometry map.

Apart from learning latent representations of textured face meshes, neural networks can be used also for the animation process itself. For example, Lombardi *et al.*[4] implement a video-driven and a control-point-based animation approach that allows animating the deep face model from three integrated cameras in a VR headset or by dragging control points of the face model. While such approaches allow direct control over the face model, they also require an experienced animator. Other approaches for facial animation predict parameters from audio signals. Cudeiro *et al.*,[9] for example, use time convolutions to predict 3-D vertex offsets for a neutral reference mesh based on DeepSpeech audio features. Similarly, Yang *et al.*[1] presents an audio-based approach for mouth animations. They use a voice conversion network to disentangle speech content and identity information. Using LSTMs, they map voice content embeddings to 2-D facial landmark positions, which allow warp-based animation of cartoon images or animating natural human images using an existing image-to-image translation network. In contrast to previously mentioned methods, we aim at performing realistic and automatic mouth animation only from semantic labels, like the paper by Paier *et al.*[10] They present an approach for visual speech animation using graph-cuts to select an optimal sequence of visual speech samples from an annotated animation database. While their approach works well in general, it still depends on the size of the animation database. For example, if certain combinations of visemes are not available, transitions between concatenated speech samples may contain artifacts (due to coarticulation), which results in an unnatural

facial animation sequence. We circumvent this problem by training an auto-regressive neural network that learns the dynamics of visual speech and is able to synthesize realistic transitions even if certain viseme combinations are not available in the training data.

Our animation method builds upon a deep neural face model, which is created from captured multiview video. This face model allows synthesizing textured face meshes from a low-dimensional latent expression vector. Using this latent representation, we train an auto-regressive animation model that synthesizes sequences of latent expression vectors, given a semantic label (which describes the target facial expression/motion). The auto-regressive approach further simplifies the resulting animation procedure, since we are able to synthesize parameter sequences without further data structures or the need for blending between concatenated sequences.

SYSTEM OVERVIEW

This section gives a high-level overview of the proposed framework, which consists of three stages that are detailed in the remainder of this document. To perform example-based animation, we create a database of relevant facial expressions from a captured multiview video footage.

In a second step, we compute an animatable face model from the captured data. This face model is based on linear blend-shapes to represent the approximate face geometry for each captured video frame. As the blend-shapes capture only rigid motion and large-scale deformations, we additionally extract dynamic face textures to represent small motions and details as well as complex areas like the oral cavity and the eyes. The reconstruction of geometry and texture is an important step, as it provides a robust way for changing head-pose and expression during animation. In order to efficiently use the extracted face performance data, we train a neural face model (i.e., a VAE) that is capable of jointly synthesizing geometry as well as texture from a low-dimensional facial expression vector.[4,10]

Using the neural face model, we represent high-dimensional geometry and texture data with a single low-dimensional latent expression vector. Apart from compressing high-dimensional data, the latent representation also allows easy sampling and interpolation of realistic facial expressions without introducing artifacts in texture or geometry.[10] While our neural face model is capable of reconstructing, sampling, and interpolating captured facial expressions, it does not capture the dynamics.

Therefore, as a third step, we train an animation network, which is capable of learning the dynamics of facial expressions. This network is trained on sequences of latent expression vectors, which we annotate with textual labels that describe the displayed viseme or the facial expression. During training, the animation network learns an embedding for each label and a rule how to predict the successive facial animation parameters from the previous parameter, a given sequence label, and a style vector. The style vector allows representing different ways of showing the same action. For example, smiling with or without visible teeth.

After training, we are able to synthesize realistic facial animations from a sequence of semantic expression labels. This enables us to:

- animate visual speech directly from text/visemes;
- perform simple and fast facial animation based on a high-level description of the content;
- capture and adjust the style of facial expressions by modifying the low-dimensional style vector.

The synthesized animation parameters are used to reconstruct sequences of textured face meshes that can be rendered with existing graphic APIs (e.g., OpenGL) or game engines.

FACIAL PERFORMANCE CAPTURE

This section presents the neural face model (i.e., a VAE), which we use in our experiments to synthesize face geometry as well as texture from a low-dimensional expression vector as in the paper by Lombardi *et al.*[4] and Paier *et al.*[5,10] We choose this method, because it yields high visual quality without the need for manual intervention. For the creation of the face model, we captured an actress in a volumetric video studio. During the capture session, we asked her to show several facial expressions like opening the mouth, smiling or being sad as well speaking in order to capture different visemes that allow synthesizing visual speech. Based on the captured data, we compute a parametric representation of the actress' face. For this purpose, we estimate rigid motion $\mathbf{T}$ and blend-shape weights $\mathbf{b}$ to describe the head pose as well as the facial expression. The head pose $\mathbf{T}$ is represented by a 6-D parameter vector, which consists of a translation vector as well as three rotation angles. PCA weight vector $\mathbf{b}$ is a 15-D column vector containing the shape weights and $\mathbf{B}$ represents the PCA-basis of facial expressions. These PCA shape weights will also be used later for training our neural face model:

$$\mathbf{x}_{\mathbf{t},\mathbf{b}} = \mathbf{T}(\mathbf{x}_0 + \mathbf{B}\mathbf{b}). \quad (1)$$

The column vector $\mathbf{x}_{\mathbf{t},\mathbf{b}}$ contains all vertex coordinates of the deformed mesh. $\mathbf{x}_0$ corresponds to the mean face geometry and $\mathbf{T}$ represents the rigid transform that is applied as a last step on all deformed vertices. Our facial performance capture pipeline uses calibrated multiview video streams and facial landmarks as input data.[11] Initially, we register the linear face model to one video frame, where the actress shows a neutral face and use it as a reference for all subsequent pose and shape estimations:

$$\mathcal{E}_{lm}(\mathbf{t},\mathbf{b}) = \sum_{c}^{C}\sum_{i} \left|\mathbf{m}_{c,i} - \hat{\mathbf{m}}_{c,i}\right|^2. \quad (2)$$

The reprojection error (2) corresponds to the projected distance between detected 2-D landmark positions ($\mathbf{m}_{c,i}$) and the corresponding location ($\hat{\mathbf{m}}_{c,i}$) on the face model:

$$\mathcal{E}_{\text{img}}(\mathbf{t},\mathbf{b}) = \sum_{c}^{C}\sum_{p\in I} \left|\mathcal{J}_c(p) - \mathcal{I}_{\mathbf{t},\mathbf{b}}(p)\right|^2 \quad (3)$$

$$\mathcal{E}_{\text{tracking}}(\mathbf{t},\mathbf{b}) = \mathcal{E}_{\text{img}} + \lambda\mathcal{E}_{lm} + \gamma|\mathbf{b}|^2. \quad (4)$$

For the initial registration, we minimize only the reprojection error (2). For all other frames, we minimize (2) as well as the intensity difference (3) between captured images and the rendered face model. The registration process is implemented as a nonlinear optimization task (4) that is solved using the Gauss–Newton method. In order to make the estimation of the shape weights $\mathbf{b}$ more stable, we include an $L2$ regularization term in the objective function (4). λ and γ are weight factors to control the influence of the landmark data term (2) and shape regularization. For more details on the face geometry tracking, please refer to the paper by Paier *et al.*[5]

Texture Extraction

The texture extraction process selects a source camera c for texturing each triangle f_i in each frame t of a captured multiview sequence. This allows the computation of the color of all texels that belong to a certain triangle f_i given the captured image of camera c, the current face geometry, and the camera calibration of c. The challenge, however, is the simultaneous selection of the optimal source camera for all triangles at all-time steps. In our case, the optimal choice maximizes the visual texture quality for each triangle and minimizes, at the same time, the visibility of seams (e.g., visible edges in the texture where adjacent triangles are textured from different cameras) and temporal artifacts (e.g., sudden changes of the texture over time caused by changing source cameras).

Therefore, we rely on a graph-cut based approach[3] that simultaneously optimizes all three terms in (5) to create a visually pleasing sequence of textures from each multiview video:

$$\begin{aligned}\mathcal{E}_{\text{tex}}(C) = &\textstyle\sum_t^T\sum_i^N \mathcal{D}(f_i^t, c_i^t)\\ &+\lambda\textstyle\sum_{i,j\in\mathcal{N}} \mathcal{V}_{i,j}(c_i^t, c_j^t)\\ &+\eta\mathcal{T}(c_i^t, c_i^{t-1}).\end{aligned} \quad (5)$$

C denotes the set of source camera IDs for all mesh triangles. The first term $\mathcal{D}(f_i, c_i)$ defines a measure for the visual quality of a triangle f_i textured by camera c_i. It uses a heuristic related to the area of f_i projected on the image plane of camera c_i.

$\mathcal{V}_{i,j}(c_i, c_j)$ represents a spatial smoothness constraint, which relates to the sum of color differences along the common edge of two triangles f_i and f_j that are textured from different cameras c_i and c_j. The last term $\mathcal{T}(c_i, c_j)$ ensures temporal smoothness by penalizing changes of the source camera c_i of a triangle f_i between consecutive time steps. Without such a term, the extracted dynamic textures are not temporally consistent, i.e., the source camera of a triangle can change arbitrarily between two consecutive texture frames, which causes temporal artifacts in the resulting texture sequence. We solve the selection of source cameras C for all mesh-triangles in all frames simultaneously using the α-expansion algorithm.[12] For more details on the dynamic texture extraction, please refer to the paper by Paier *et al.*[3]

The advantage of this approach is that we generate high-quality textures even with an approximate geometry model. Since each triangle is textured only from a single source camera, our approach does not suffer from blur or ghosting artifacts. Moreover, the extracted dynamic textures capture all information that is not represented by geometry like fine motions and deformations, changes in texture, and occlusions/dis-occlusions (e.g., eyes and oral cavity). Together, dynamic geometry and texture represent almost the full appearance of the actress' face in each frame.

Neural Face Representation

The result of the facial performance capture is a sequence of textured meshes. Each frame is represented by a rigid motion $\mathbf{T}$, blend-shape weights $\mathbf{b}$, and an RGB image as texture. In order to use these

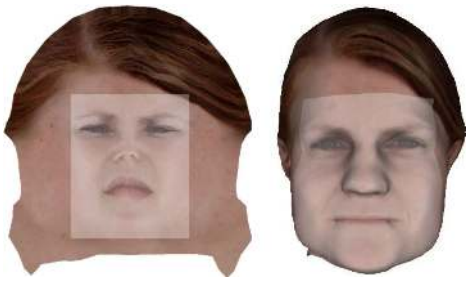

FIGURE 2. Animated region of interest. We select the modified area in texture space by defining a rectangle.

textured mesh sequences in a learning-based animation scheme, we train a deep generative face model[5] that is capable of reconstructing PCA weights and face textures from a low-dimensional parameter vector. To simplify the training, we select a region of interest in texture space (see Figure 2) that will be represented by the neural face model. Figure 3 shows the architecture of our neural face model based on a VAE.[6] The auto-encoder receives PCA face shape weights as well as the textures from the performance capture stage as input and transforms them into a 1024-D latent representation of the facial expression. Based on the latent expression vector, we reconstruct face shape weights as well as texture. We use an adversarial training strategy based on a discriminator network, which is trained simultaneously with the auto-encoder. The purpose of the discriminator is classifying, whether a texture is synthesized or real. By incorporating the classification error during training, we can improve the visual quality of reconstructed textures. The full training objective function consists of the absolute difference between predicted texture and target texture, the adversarial texture loss and the mean-squared error between the predicted PCA weights and the target PCA weights. For a more detailed presentation of our neural facial model please refer to the papers by Paier *et al.*[5,10]

ANIMATION AND RENDERING

After training the neural face model, we represent the reconstructed meshes and dynamic textures with sequences of latent expression vectors. Additionally, we annotate these sequences by adding semantic labels describing the presented content. We annotate general facial expressions as well as speech. Table 1 describes all 13 visemes that we use for annotating speech. General facial expressions are marked with one of the six following labels: <openmouth>, <sad>, <smile>, <disgust>, <blowcheeks>, and <idle>. With the annotated data, we train an auto-regressive network that enables us to synthesize sequences of latent expression vectors given a single label or a sequence of labels. The generated latent expression vectors are then used to reconstruct animated and textured meshes, which can be rendered with common graphic APIs or game-engines.

In order to synthesize animation parameters, the network receives the last animation parameter/timestamp as well as the current sequence label, the next sequence label, and a style vector. Providing the current and the next sequence label helps resolving ambiguities when animating visual speech, because the appearance of visemes does not only depend on the current viseme itself but also on its neighbors (see Figure 5). This is caused by an effect called coarticulation, which refers to a phenomenon in speech when two successive sounds are articulated together.

Similar effects can occur when animating general facial expressions, as they can be performed in many different ways (e.g., smiling with or without showing the teeth, Figure 8). Therefore, we condition our network on a style vector that is learned automatically during training. Label and style are zero-based indices that are transformed to a 32-D feature vector by separate embedding layers. We use 19 different feature vectors to represent labels and 514 different style vectors (one style vector for each annotated sequence).

We follow a similar approach as Cuderio *et al.*[9] who condition their network on a subject ID to represent the speaking styles of different people. However, since we want to learn different styles of the same action, we have to estimate a separate style vector for each annotated training sequence, as we can only assume that style remains constant within the sequence. Since this assumption is less restrictive as in the paper by Cuderio *et al.*,[9] we have to further constrain the learning of style vectors by keeping the style feature dimensionality as low as possible. Moreover, we encourage the network to make full use of information that is stored in labels by reconstructing animation parameters not only from the final feature map but also from the feature map, which is produced by the second residual block (i.e., before introducing style information). Therefore, style embedding and the last residual block are only responsible for reconstructing a small expression-residual that represents different styles of the same expression (e.g., smiling with or without showing teeth). This forces the network to generate plausible animation parameters also without style information and, hence, prevents the network from memorizing expression parameters only through style information. Figure 4 shows the architecture of our animation network. The network is based on a simple residual architecture.

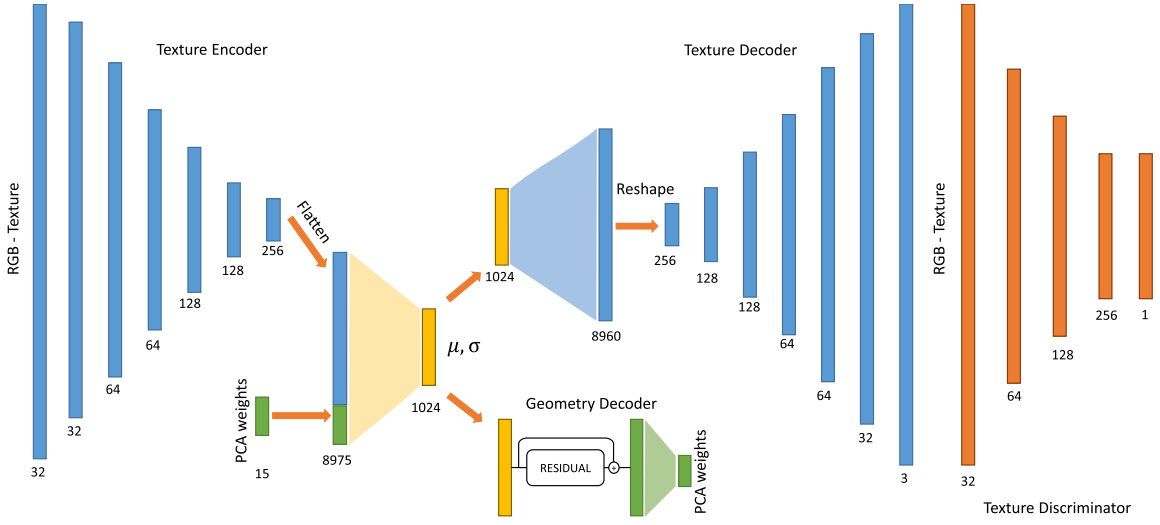

FIGURE 3. Network architecture of the neural face model. The network consists of five parts: a convolutional texture encoder/decoder (blue), a geometry decoder (green), a fully connected bottleneck (yellow) that combines information of texture and geometry into a latent code vector (μ), and deviation (σ). A texture discriminator network (orange) classifies textures as real or as synthetic. This figure is taken from the paper by Paier *et al.*[10]

Each residual block consists of two fully connected layers (linear layer, batch-norm, leaky-ReLU) that compute a residual, which is added to the input feature vector. A time-stamp is represented by a scalar between 0 and 1. A value equal to 0 marks the beginning of the sequence, and a value equal or above 1 marks the end of the sequence. As this network considers previous animation parameters as well, the resulting animation stream is free of jumps/artifacts, and we do not need to perform any kind of additional blending between consecutive sequences. We use a batch-size of 32, a leakiness of 0.01 for all ReLUs in our network, and use the Adam optimizer with default parameters to train our network for approximately 175 epochs with an initial learning rate of 0.005 and exponential learning rate scheduling ($\gamma = 0.95$).

TABLE 1. Viseme dictionary.

Phoneme (CELEX)	Viseme
b m p	P
d n s t	T
@ N g h k x	-
l	L
f v	F
I i j	I
E e	E
a	A
& O Q o	O
U Y u y	U
r	R
z	S
S	$

After training, we are able to synthesize an animation sequence according to a given label and style by iteratively evaluating the network with the previous animation parameter, time-stamp, and the desired label as well as style parameters. The last parameter and time-stamp are stored in separate variables that are updated after each network evaluation. While we initialize the last parameter vector only once with zeros, we have to reset the time-stamp to zero every time after finishing the parameter generation for an expression label. Using our neural face model, we are able to reconstruct face geometry and textures from the synthesized animation parameter sequences. The animated face model is then rendered and displayed with a standard graphics pipeline (e.g., OpenGL).

EXPERIMENTAL RESULTS AND DISCUSSION

This section presents still images of the proposed animation technique, while an accompanying video can be found in the supplementary material. Figure 9

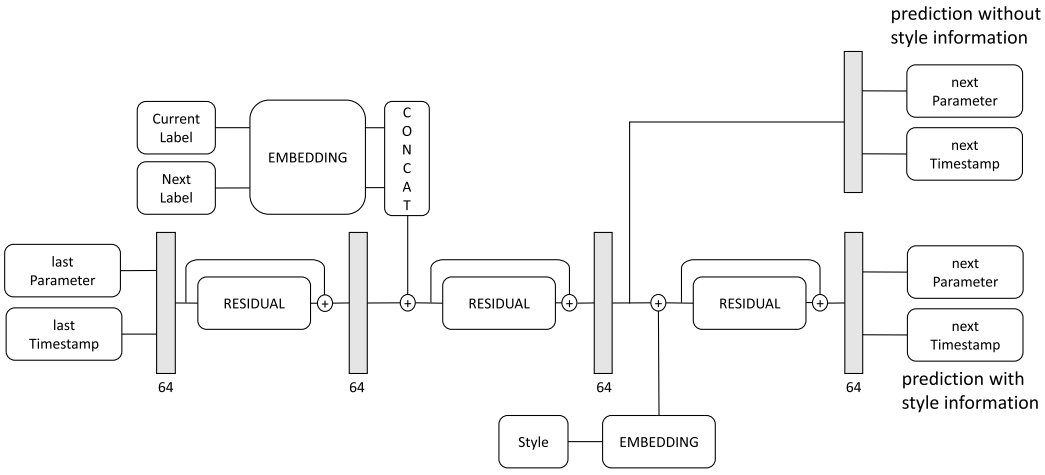

FIGURE 4. Animation network architecture. It transforms previous animation parameters and time-stamps into a 64-D feature map. After the first residual block, we integrate label information followed by another residual block. Using a linear layer, we reconstruct the next animation parameter and time-stamp from the resulting feature map. Moreover, we add style information to the feature map that allows the reconstruction of similar facial expressions that only differ in style. After adding style information, we process the resulting feature map with a third residual block and reconstruct the next animation parameter as well as the time-stamp.

shows synthesized visual speech as well as general facial expressions.

For our experiments, we captured a German-speaking actress with 16 synchronized video camera pairs that were equally placed around her. We captured different facial expressions as well as speech. We asked her to present single words and short sentences that are related to weather forecasts. We annotated facial expression and viseme sequences manually. The neural face model was trained with approximately 9200 frames. The animation network was trained with approximately 3500 frames. The effective capturing resolution for the head is approximately 520 $\times$ 360 pixel. Four camera pairs were located on the ground, eight camera pairs were placed at the eye level, and four more camera pairs were placed above the actress.

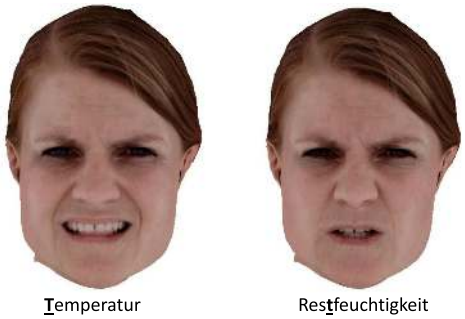

FIGURE 5. Visually different examples of German visemes for the letter "*t*."

Additionally, we captured static facial expressions with 14 synchronized D-SLR cameras (Canon 550D) that were placed around the actress as well. Based on these still images and the proposed method in the paper by Paier *et al.*,[5] we compute the animatable head geometry model for 3-D pose and expression tracking. While using high-quality hardware during data acquisition, our method is not restricted to this exact configuration. A low-cost setup could consist of three video cameras that capture the subject's face from left, right, and frontally. Figure 1 demonstrates the advantages of our neural face representation as it can realistically reproduce different facial expressions, for example, with almost closed mouth, opened mouth, visible tongue, and teeth. The simple geometry accounts only for large-scale deformations (i.e., the jaw movement), while details like tongue or teeth are captured in texture space (see Figure 1, left column). Figure 6 shows the evolution of the parameter prediction error on the training (blue) and test (orange) set. The optimization is stopped after approximately 175 epochs, since the error does not decrease anymore. The final training and test errors (RMSE) are 0.28 and 0.35, respectively. Target animation parameters are normalized to have zero mean and a standard deviation of 1. The test set consists of complete sequences that have been randomly selected (approx. 25% of the dataset). All preprocessing steps, network training,

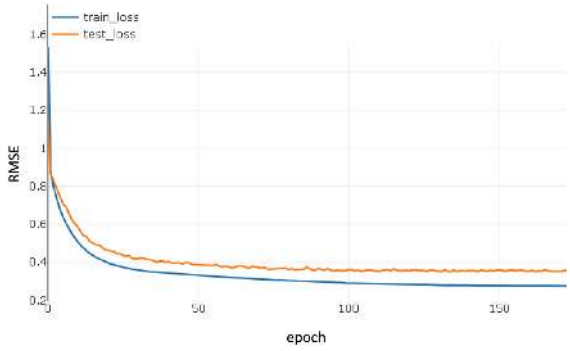

FIGURE 6. Evolution of parameter and time-stamp prediction error on the training (blue) and test set (orange). After approximately 175 epochs, the loss does not improve anymore and we stop optimization.

and experiments have been carried out on a regular desktop computer with 64 GB Ram, 2.6 Ghz CPU (14 cores with hyperthreading), and one GeForce RTX2070 graphics card. Animation and rendering runs at a rate of approximately 40 fps.

Figure 7 demonstrates the capability of our animation network to reproduce effects like coarticulation, where the appearance of visemes depends on the neighboring visemes as well. Therefore, we synthesized the German word Italien (Italy) in two different ways. First (middle row), we provide the current and the next viseme label, while the second version (bottom row) was synthesized with an alternative network architecture that receives only the current viseme label. The animation in the middle row is clearer, more expressive, and visually closer to ground truth (top row). This is most probably caused by the fact that providing only the current viseme forces the animation network to learn the average appearance/dynamics of each viseme, which results in a less natural pronunciation.

Figure 9 shows that the proposed system is capable of producing correct mouth expressions according to different label sequences. The upper two examples show an animated face while speaking words that are not part of the training database. For the synthesis of visual speech, we use a zero style vector. While the resulting animation deviates slightly from the ground-truth, the produced visemes are clear and well recognizable.

To demonstrate the animation of general facial expressions, we produced another facial animation sequence that starts with a neutral expression, changes to smiling, and goes back to neutral. While a

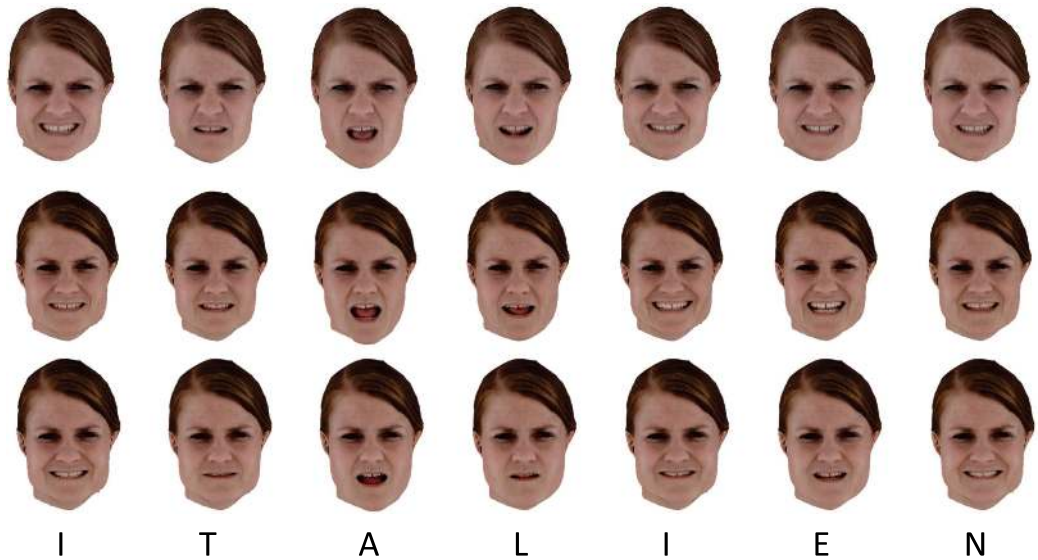

FIGURE 7. Synthesized visemes for the animation of the German word Italien (Italy). The upper row shows the ground truth. The middle row was generated using the labels of the current and the next viseme, while the bottom row was synthesized based only on the current viseme. The animation in the middle row is clearer and more expressive. This demonstrates another important aspect of visual speech production, which is caused by coarticulation. The appearance of visemes is not only determined by the viseme itself, but also its neighbors. Knowing the next viseme/label allows the network to adapt the appearance accordingly.

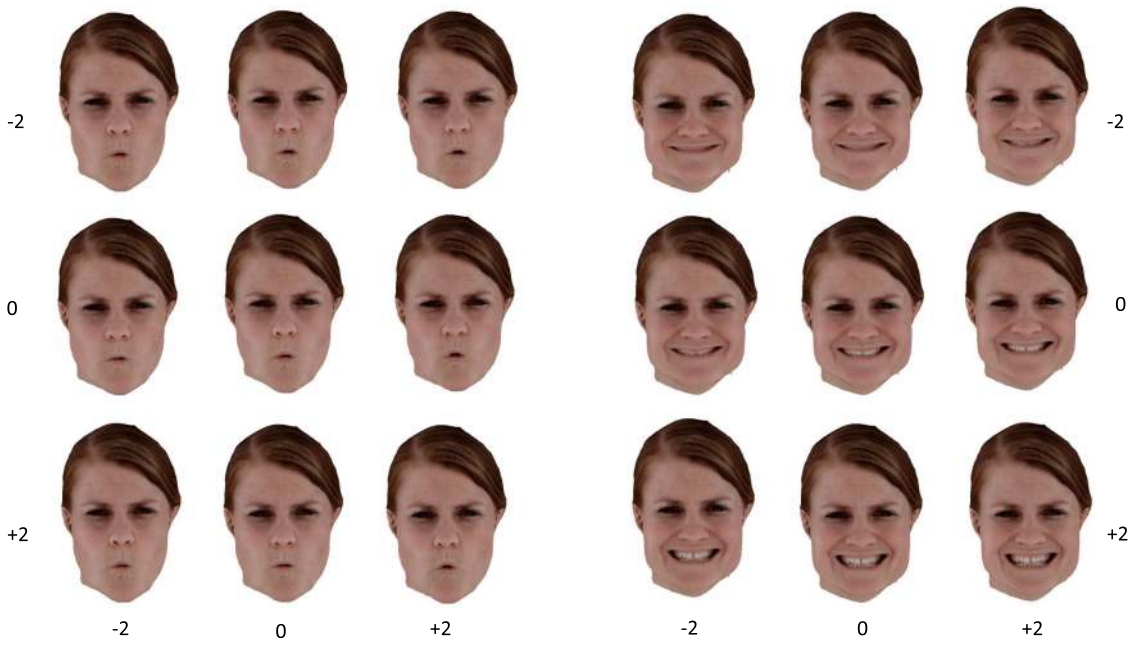

FIGURE 8. Impact of the learned style parameter. We show nine versions of the viseme "O" and the smile expression. Columns and rows correspond to the first and second dimension of a 2-D style vector. For this animation, we use styles coordinate values between -2 and $+2$. All renderings show the facial expression at the middle of the animation sequence.

zero style vector results in a smile with visible teeth (Style B), we can modify the appearance of this expression by only changing the style vector, which creates a different smile sequence, where teeth are not visible anymore (Style A). Figure 8 demonstrates the capability of our system to modify the style of an animated facial expression, while keeping the expression label constant. For this experiment, we trained our animation network with 2-D style vectors to capture differences in appearance that cannot be explained by semantic labels alone. The learned style vectors are regularized such that distribution of all style vectors corresponds to a standard normal distribution. We visualize the impact of the style parameter, by varying the style coordinate values between -2 and $+2$ and render the facial expression at the middle of the animated sequence. The results show that we can successfully modify the style of the animation sequence, while keeping the semantics. This enables, for example, a human editor to modify a generated animation sequence in a postprocessing step by simply fine-tuning a 1-D or 2-D style parameter.

While our system is, in general, able to synthesize realistic animation sequences based only on semantic labels, there are certain limitations as well. For example, our architecture is able to model short-term relations in the input sequence since we provide the current and the following semantic label. This is sufficient to capture effects like coarticulation, but it does not allow modeling long-term relations (e.g., processing a complete sentence), which could be helpful to predict additional animation parameters for eyes, the overall facial expression/emotion or global motions like head movement. Apart from that, our animation model predicts all parameters from a static semantic label without the need for further input. While this simplifies the animation-process, it also reduces the ability to control, timings, or duration of an animation sequence. However, it is possible to specify a speed-factor, which can be used during the auto-regressive decoding to scale the animation-time. A speed-factor below 1 would result in a longer animation sequence, while a value above 1 would result in faster animation.

CONCLUSION

We present a new method for example-based facial animation using auto-regressive neural networks. Our approach is based on a high-quality neural face model that can reconstruct realistic facial expressions from a low-dimensional latent feature

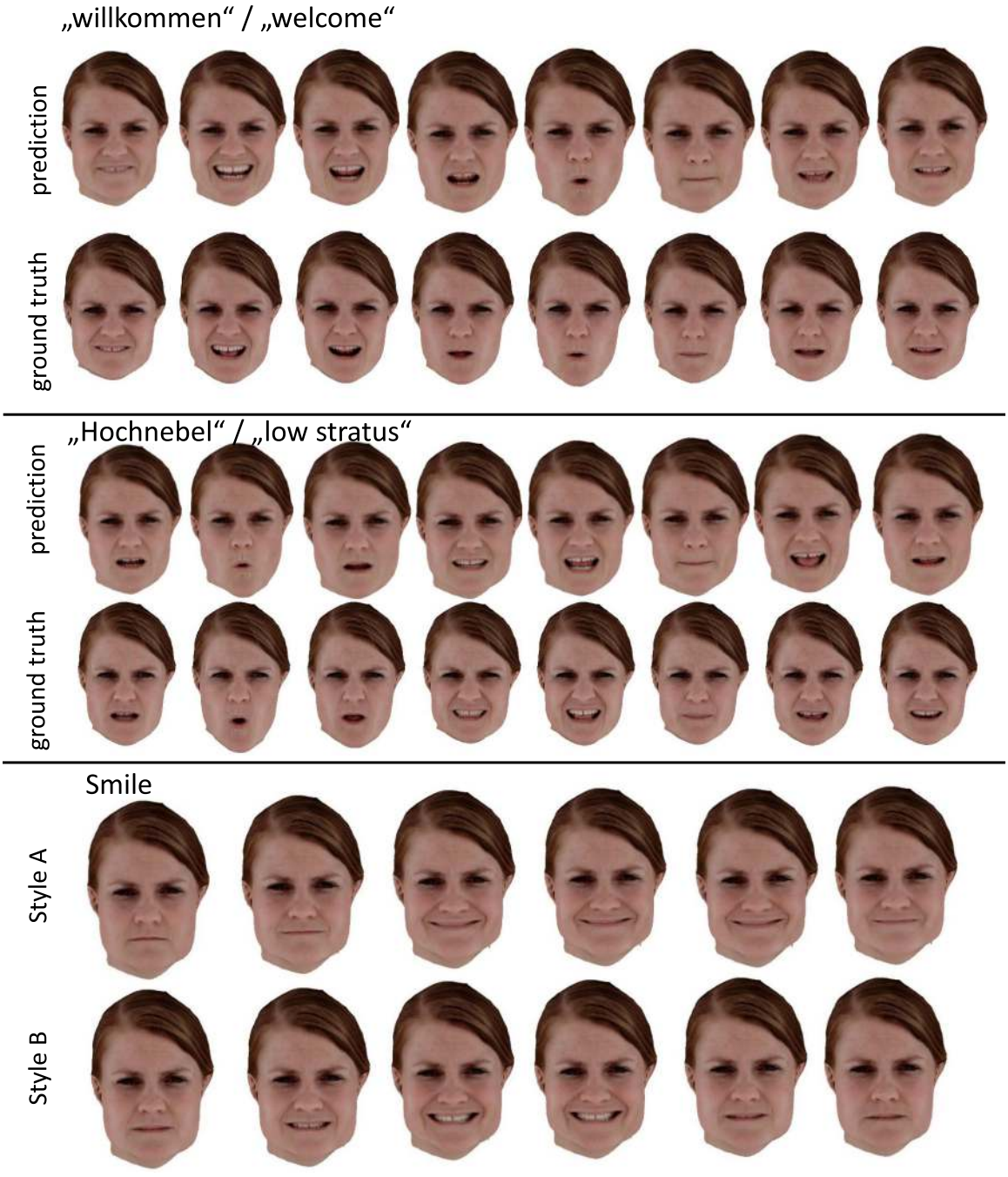

FIGURE 9. Synthesized visual speech as well as a general facial expressions. The two upper examples show an image sequence for two German words that are not part of the training data. Each face picture represents one viseme. The third example shows a synthesized smile sequence with two different styles that appear in training data.

vector. Based on an annotated database of dynamic facial expressions, we train an auto-regressive network that successfully learns the dynamics of facial expressions. After training, we are able to synthesize realistic facial animations from a sequence of semantic labels that act as a high-level descriptor of the target content. Moreover, we present a robust approach that disentangles style and content, which enables capturing and reproducing facial expressions (with the same semantic label) that

differ only in style, for example, smiling with or without showing the teeth.

ACKNOWLEDGMENTS

This work was supported in part by the European Union's Horizon 2020 research and innovation programme under Grant 762021 (Content4All) and Grant 952147 (Invictus). This article has supplementary downloadable material available at https://doi.org/10.1109/MCG.2021.3068035, provided by the authors.

REFERENCES

1. Y. Zhou, X. Han, E. Shechtman, J. Echevarria, E. Kalogerakis, and D. Li, "Makeittalk: Speaker-aware talking-head animation," *ACM Trans. Graph.*, vol. 39, no. 6, pp. 1–15, 2020, doi: 10.1145/3414685.3417774.
2. J. Thies, M. Zollhöfer, M. Nieûner, L. Valgaerts, M. Stamminger, and C. Theobalt, "Real-time expression transfer for facial reenactment," *ACM Trans. Graph.*, vol. 34, no. 6, 2015, Art. no. 183, doi: 10.1145/2816795.2818056.
3. W. Paier, M. Kettern, A. Hilsmann, and P. Eisert, "A hybrid approach for facial performance analysis and editing," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 4, pp. 784–797, Apr. 2017. [Online]. Available: https://doi.org/10.1109/TCSVT.2016.2610078
4. S. Lombardi, J. M. Saragih, T. Simon, and Y. Sheikh, "Deep appearance models for face rendering," *ACM Trans. Graph.*, vol. 37, no. 4, Jul. 2018, Art. no. 68, doi: 10.1145/3197517.3201401.
5. W. Paier, A. Hilsmann, and P. Eisert, "Interactive facial animation with deep neural networks," *IET Comput. Vis., Special Issue Comput. Vis. Creative Industries*, vol. 14, no. 6, pp. 359–369, Sep. 2020, doi: 10.1049/iet-cvi.2019.0790.
6. D. Kingma and M. Wellington, "Auto-encoding variational Bayes," in *Proc. 2nd Int. Conf. Learn. Representations*, 2014, pp. 1–14. [Online]. Available: https://arxiv.org/abs/1312.6114
7. R. Li *et al.*, "Learning formation of physically-based face attributes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 3407–3416, doi: 10.1109/CVPR42600.2020.00347.
8. P. Chandran, D. Bradley, M. Gross, and T. Beeler, "Semantic deep face models," in *Proc. Int. Conf. 3D Vis.*, 2020, pp. 345–354, doi: 10.1109/3DV50981.2020.00044.
9. D. Cudeiro, T. Bolkart, C. Laidlaw, A. Ranjan, and M. Black, "Capture, learning, and synthesis of 3D speaking styles," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 10101–10111, doi: 10.1109/CVPR.2019.01034.
10. W. Paier, A. Hilsmann, and P. Eisert, "Neural face models for example-based visual speech synthesis," in *Proc. 17th Eur. Conf. Vis. Media Prod., Virtual Event*, Dec. 2020, pp. 1–14, doi: 10.1145/3429341.3429356.
11. V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1867–1874, doi: 10.1109/CVPR.2014.241.
12. Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1222–1239, Nov. 2001. [Online]. Available: https://doi.org/10.1109/34.969114.

WOLFGANG PAIER is currently a Research Associate with Fraunhofer HHI, Berlin, Germany. His research interests include 3-D scene analysis, and image- as well as learning-based approaches for facial performance capture, animation, and editing. He received the B.S. degree and the M.Sc. degree in computer science from the Technical University of Graz, Graz, Austria, in 2009 and the Free University of Berlin, Berlin, Germany, in 2013, respectively. He is the corresponding author of this article. Contact him at wolfgang.paier@hhi.fraunhofer.de

ANNA HILSMANN is currently heading the Computer Vision & Graphics Group, Fraunhofer HHI, Berlin, Germany. Her main research interests cover 3-D image and video analysis, model-based deformable tracking and 3-D reconstruction, as well as image- and video-based rendering, animation, and editing. She received the Dipl.-Ing. degree in electrical engineering and information technology from RWTH Aachen, Aachen, Germany, in 2006 and the Dr.-Ing. degree in computer science from HU Berlin, Berlin, Germany, in 2014. Contact her at anna.hilsmann@hhi.fraunhofer.de.

PETER EISERT is currently a Professor of visual computing with Humboldt University, Berlin, Germany and heads the Vision & Imaging Technologies Department, Fraunhofer HHI, Berlin, Germany. He has authored or coauthored more than 200 conference and journal papers and is an Associate Editor of the *International Journal of Image and Video Processing* as well as on the editorial board of the *Journal of Visual Communication and Image Representation*. His research interests include 3-D image analysis and synthesis, face processing, image-based rendering, deep learning, computer vision, and computer graphics. He received the Dr.-Ing. degree from the University Erlangen, Erlangen, Germany, in 2000 and was a Postdoctoral Fellow with Stanford University. Contact him at peter.eisert@hhi.fraunhofer.de.