Front.Comput.Sci.
DOI

RESEARCH ARTICLE

Speech-Driven Facial Animation with Spectral Gathering and Temporal Attention

Yujin Chai[1], **Yanlin Weng** (✉)[1], **Lvdi Wang**[2], **Kun Zhou**[1]

1 State Key Lab of CAD&CG, Zhejiang University, Hangzhou 310058, China
2 FaceUnity Technology Inc., Hangzhou 310011, China

Abstract In this paper, we present an efficient algorithm that generates lip-synchronized facial animation from a given vocal audio clip. By combining spectral-dimensional bidirectional long short-term memory and temporal attention mechanism, we design a light-weight speech encoder that learns useful and robust vocal features from the input audio without resorting to pre-trained speech recognition modules or large training data. To learn subject-independent facial motion, we use deformation gradients as the internal representation, which allows nuanced local motions to be better synthesized than using vertex offsets. Compared with state-of-the-art automatic-speech-recognition-based methods, our model is much smaller but achieves similar robustness and quality most of the time, and noticeably better results in certain challenging cases.

Keywords speech-driven facial animation, spectral-dimensional bidirectional long short-term memory, temporal attention, deformation gradients

1 Introduction

Recent years have witnessed an accelerating evolution of facial animation technologies. Animating a 3D face avatar, for example, is no longer the privilege of those equipped with professional skills and devices. Thanks to some of the exciting advances in the field, anyone with a smartphone can control the facial expression of a vivid virtual avatar in real time, with nothing but their *own face* [1,2].

To push the limit further, people naturally wondered whether an excerpt of vocal recording or a transcript would be sufficient to deduce a corresponding facial animation, as a driving face might be absent in certain scenarios. Such examples may include a virtual assistant whose utterances are synthesized on the fly. It turns out that some surprisingly faithful results can be achieved [3–7].

However, while face-driven virtual avatars (such as Apple's Animoji and Memoji) have already been entertaining world-wide users for a while, we still need to solve at least two challenges before *speech-driven* facial animation techniques can reach that same maturity: *robust audio processing* and *effortless motion retargeting*. Together they would allow an end-to-end system to generalize well to both unheard voices and unseen avatars—an ability vital to consumer-facing products.

1.1 Vocal audio processing

Human beings have an elaborate articulatory system. Producing speech sounds is largely a deterministic process that involves not only the visible interactions between facial muscles and skin tissues, but also nuanced coordination across multiple internal organs. On the other hand, inferring the facial motion that produces a given speech is an inherently ambiguous reverse process [3, 4]. Furthermore, real-world speech signals vary tremendously due to the speaker's age, gender, emotional state, the environment, and the recording devices [8]. Previous works based on deep neural networks [9, 10] can handle such ambiguities and variations to some extent, provided that sufficient training data is available.

Received March 31, 2020; accepted September 23, 2020

E-mail: weng@cad.zju.edu.cn

While a high-quality, pre-trained automatic speech recognition (ASR) module can be readily used to extract robust speech features [6, 11], it imposes a significant overhead to the model complexity and runtime performance.

Our insight here is that mapping speech to facial animation is different enough from common speech recognition tasks that a feature extractor designed for the latter may not be the ideal choice for the former. We thus propose a new *speech encoder* that is tailored to our specific task. Unlike a fully-convolutional network [4], it employs a bidirectional long short-term memory (LSTM) [12] network *along spectral dimension* which can better capture long distance correlations of formants in the mel spectrogram. We further introduce a *temporal* attention mechanism that allows the model to focus on the few audio frames that are influential to the facial motion but otherwise easy to miss.

The proposed encoder network compares favorably against the state-of-the-art [6] built on a pre-trained ASR module [13] in terms of accuracy and robustness, but is only *a fraction* of its size.

1.2 Motion representation

The other key challenge in speech-driven facial animation lies in the way facial motions are represented. Being able to effortlessly drive an arbitrary 3D face model, even one with a drastically different mesh topology than those seen and learned by the model, would be highly desirable. Two popular choices in this regard are *low-dimensional expression coefficients* (*e.g.*, blendshapes) [5, 11, 14–17] and *per-vertex offsets* from a globally aligned expressionless template mesh [4, 6]. But a potential limitation shared by both representations is that the face models used for training and inference must have exactly the same *underlying structure*: same rig, same blendshape bases, or same mesh tessellation.

Inspired by the study of deformation transfer [18–20], we let our model's decoder network output *deformation gradients* as an intermediate representation of the target facial motion, from which we can reconstruct the facial mesh either using the original template or a new 3D face, possibly with a different topology.

1.3 Contributions

The key technical contributions of our approach include:

1. A lightweight, robust speech encoder designed specifically for the task of animating 3D face avatars from input vocal audio.
2. Introducing deformation gradients as the motion representation for better handling of non-rigidity and easier generalization to topologically different faces.

Based on these ideas, we present an end-to-end speech-driven facial animation algorithm (**Fig. 1**). It outperforms state-of-the-art methods [4, 6] in several challenging cases (*e.g.*, those involving *lip closures* or *lasting vowels*). Without using any pre-trained ASR module, our model is compact and runs in real time with low latency. Once trained, it generalizes robustly to unheard voices and unseen face models. To assess the quality of speech animations generated by our algorithm, please watch the supplementary video[1)].

2 Related works

We briefly review the prior arts those are most pertinent to the task of generating facial animations from audio. For the broader topic of facial capture and dynamic manipulation, we refer the readers to the survey by Orvalho *et al*. [21].

Procedural methods map phonemes in the audio to visemes following certain predefined rules. One of the main challenges is how to realistically handle coarticulations [22–24]. Cohen and Massaro [25] propose dominance functions to evaluate degree of a certain viseme in a given context. Xu *et al*. [26] use phone bigrams to handle coarticulation. But it is difficult to cover all possible coarticulation cases in real-world speech. Edwards *et al*. [3] propose a jaw-lip action model with an emphasis on artistic control.

Bregler *et al*. [27] proposes an **example-based method** to rewrite video frames to match a new audio clip via automatic mouth tracking and image warping. Ezzat *et al*. [28] map the phonemes into clustered principal component analysis (PCA) coefficients that represent the shape and texture of the lower face. Taylor *et al*. [29] use active appearance model (AAM) to model variants in shapes and textures of the lower face, and match variable-length phoneme substrings with similar appearances into dynamic visemes.

Brand [30] estimates a **hidden Markov model** (HMM) from facial landmarks in the video and synthesize the most probable sequence through trajectory optimization. Xie and Liu [31] model the movements of articulators based on dynamic Bayesian networks. Wang *et al*. [32] map mel-frequency cepstral coefficients (MFCCs) to PCA coefficients with an HMM, which is further extended by Zhang *et al*. [33] with context-dependent deep neural network hidden Markov

[1)] Also available at: `https://chaiyujin.github.io/sdfa`

model (CD-DNN-HMM) for more robust audio feature extraction.

Recurrent neural networks (RNNs) and their variants have been exploited by many [9, 14, 15, 17, 34–38] due to the sequential nature of audio and visual data. In order to take both past and future context into account, Fan *et al*. [35] adopt a bidirectional long short-term memory (BiLSTM), whereas the dependency over all time frames prevents their method from running in real-time. Suwajanakorn *et al*. [9] propose a time-delay long short-term memory to look only at the short-term future. Notably, a study conducted by Websdale *et al*. [39] shows that at least 70 ms of look-ahead is necessary in order to synthesize plausible coarticulations. Schwartz and Savariaux [40] discuss about asynchrony between visual and auditory events and prove there is a typical range of 30~50 ms auditory lead to 170~200 ms visual lead caused by many cases like preparatory lip gestures. The range should be covered by input audio context to handle such kinds of asynchronous events.

Taylor *et al*. [10] introduce a **sliding window** method that maps overlapping windows of phoneme subsequences to per-frame AAM parameters using a deep neural network. Karras *et al*. [4] adopt an encoder-decoder architecture, where a two-phase **convolutional neural network** (CNN) performs (feature-dimensional) formant analysis and (temporal) articulation analysis over sliding window of linear predictive coding (LPC) feature. Fully connected layers are then used to decode the mesh vertex offsets of the frame central to the sliding window. Following this idea, Pham *et al*. [5] and Tzirakis *et al*. [16] both choose to replace fully connected layers with RNNs to decode blendshape coefficients of template face rigs. Hati *et al*. [7] prependly attach a text-to-speech module powered by Tacotron2 [41] and WaveGlow [42] to a similar CNN-based architecture to generate speech and facial animation simultaneously from text.

Cudeiro *et al*. [6] present the impressive VOCASET dataset along with two notable ideas. First, by conditioning on speaker labels they are able to decouple motion and speaking style from face shapes. Second, by integrating a **pre-trained ASR module**, DeepSpeech [13], the audio feature extraction becomes much more robust. Our model also utilizes speaker labels to distinguish idiosyncratic styles, but replaces the DeepSpeech module by a carefully tailored speech encoding network that offers comparable robustness while being *much* smaller.

In a recent work, Tian *et al*. [15] also combine a BiLSTM **with attention mechanism** to generate facial animation from audio. There are two contrasting differences between their method and the one proposed in this paper. In their pipeline, windowed audio features are *flattened* and fed directly to the stateful BiLSTM, on which attention mechanism keeps track of the *entire history*. On the other hand, our speech encoder processes the window *in two orthogonal phases*, one frequency-wise (using BiLSTM), the other frame-wise (using attention) and our attention mechanism only focuses on the *given window*, which reduces the difficulty of training a robust attention module compared to the entire history. Furthermore, Tian *et al*. [15]'s pipeline outputs blendshape coefficients of a predefined face rigs, limiting its ability to animate unrigged avatars.

Note that Karras *et al*. [4] also demonstrate a result where the facial motion is transferred from a known face model to a new one using **deformation gradients** [18] as a post-processing step. Our use of deformation gradients is different in that we integrate it as the decoding network's direct output so that the decoupled motion can be immediately applied to an arbitrary face mesh.

A **generative adversarial networks** (GANs) based method is proposed by Vougioukas *et al*. [43] to generate realistic talking head video from audio and one still face image. A frame discriminator, determining the reality of each single frame, and two temporal discriminators, determining the reality and synchronization of video respectively, are used. Whereas, Chen *et al*. [44] hierarchically regress landmarks and generate video from landmarks. Attention-based dynamic pixel-wise loss is proposed to address pixel jittering in audiovisual-non-correlated regions. A novel regression discriminator is proposed to determine the reality of entire video and the accuracy of landmarks for each frame. These GANs-based papers tackle the speech-to-facial animation problem in the *image space*, different from 3D mesh-based approaches like Cudeiro *et al*. [6] and ours.

3 Method

Our overall algorithm follows an encoder-decoder architecture [4–6, 16]. The raw input audio sequence is processed by a sliding window. Signals within a window is converted into mel spectrogram (§ 3.1) before being fed to a three-stage deep neural network. In the first stage (§ 3.2), we perform formant analysis in the spectral dimension with bidirectional long short-term memory. In the second stage (§ 3.3), temporal transitions are aggregated with a frame-wise attention mechanism to yield a robust encoding of the windowed audio signal. The third stage (§ 3.4), controlled by the one-hot

subject label [6], follows to decode the facial motion in the form of deformation gradients [18]. Finally (§ 3.5), the deformation gradients, together with a static template mesh, are combined to reconstruct the output facial mesh corresponding to the center frame of the temporal window. **Fig. 1** illustrates the entire pipeline.

3.1 Audio preprocessing

We first convert the raw audio into spectrogram frames using short-time Fourier transform. Frames each has a duration of FFT_{win} and are separated by FFT_{hop}. To calculate the input window of mel spectrogram, we use a window of L frames and F mel-frequency bins. We further stack the first and second temporal derivatives as auxiliary features, resulting in a final tensor of shape $3 \times F \times L$.

3.2 Formant analysis with spectral gathering

Most CNN-based methods [5,16] treat (mel) spectrograms as plain images. However, as noted by Abdel-Hamid *et al*. [45], standard CNN kernels may not be suitable when handling the spectral domain as signals in different frequency bands may behave quite differently. Simply using a large kernel size is likely to cause overfitting due to the prevalence of unimportant partials.

Motivated by the successful application of spectral-dimensional long short-term memory in ASR [46] and pitch tracking [47] tasks, we propose a hybrid network architecture (**Table 1**): the mel spectrogram feature ($3\times F\times L$) is fed to two 2D convolution layers with kernel size 3×1, each followed by a max pooling with stride 2×1 along the spectral dimension to detect simple local features and again a 1×1 convolution, producing an output of dimensions $C_{\text{conv}} \times \frac{F}{4} \times L$. A *spectral-dimensional* bidirectional long short-term memory (Spec-BiLSTM) is then applied, effectively *gathering* information in the spectral dimension.

Finally, the sequence of output at all frequency bands are stacked and consumed by a fully connected layer, yielding a spectral feature $\mathbf{z}_{\text{spec}}$ ($C_{\text{spec}}\times 1\times L$). Its shape can be squeezed into $C_{\text{spec}} \times L$, as the size of spectral dimension is 1.

3.3 Articulation analysis with temporal attention

When analyzing the phonemes in the speech, *vowels* can usually be recognized by their distinct formant patterns. *Consonants*, on the other hand, are identified by the transition of formants between adjacent vowels.

We propose an attention-based [48, 49] articulation analysis network that replaces the convolution layers commonly found in state-of-the-art works [4–6, 16]. As shown in **Fig. 2** and **Table 2**, the spectral feature $\mathbf{z}_{\text{spec}}$ from the previous formant analyzer is fed to two *temporal* bidirectional long short-term memories (Time-BiLSTMs) to get a memory $\mathbf{m}$ with shape $C_{\text{time}} \times L$. This step makes sure that each frame has some knowledge about its context. Content-based attention proposed by Bahdanau *et al*. [49] is then used to decide the weight of each time frame. The central K_{qry} frames of $\mathbf{m}$ are processed by a 1D convolution operator with kernel size K_{qry} and projected linearly to get the query term $\mathbf{q}_{\text{att}}$ with shape $C_{\text{att}} \times 1$. The memory $\mathbf{m}$ is also projected linearly to get the key term $\mathbf{k}_{\text{att}}$ with shape $C_{\text{att}} \times L$. The $\mathbf{q}_{\text{att}}$ is repeated and added element-wise to $\mathbf{k}_{\text{att}}$. We use a tanh activation and project the summed array into a shape of $1 \times L$ as per-frame scores. After a softmax normalization along time frames, we can get the weights.

The final output, $\mathbf{z}_{\text{att}}$ with shape $C_{\text{time}} \times 1$, is the weighted sum of $\mathbf{m}$ along the temporal dimension. Its shape can be squeezed into C_{time}, as the size of temporal dimension is 1.

3.4 Motion decoding with deformation gradients

When representing the 3D deformation caused by facial motion, a common choice is *vertex offsets* [4, 6], *i.e*., the per-vertex displacements of a deformed facial mesh in frame t in comparison with a static expressionless template mesh. However, due to the complex non-linearity of human faces, it is difficult to single out "shape-independent" vertex offsets, even with the help of conditioning on the speaker label during training [6] (which nevertheless helps the model *learn* motion patterns across multiple speakers).

Instead of focusing on vertices directly, we adopt *deformation gradients* [18] as a local descriptor for the non-rigid deformation between the expressionless template mesh and one in motion. More concretely, let $v_i^{(k)}$ and $\tilde{v}_i^{(k)}$, $k \in 1, 2, 3$, denote the three vertices of the i-th triangle in the expressionless template and deformed mesh, respectively. To handle the deformation perpendicular to the triangle, we also compute a fourth vertex $v_i^{(4)}$ as:

$$v_i^{(4)} = v_i^{(1)} + \frac{(v_i^{(2)} - v_i^{(1)}) \times (v_i^{(3)} - v_i^{(1)})}{\sqrt{\left|(v_i^{(2)} - v_i^{(1)}) \times (v_i^{(3)} - v_i^{(1)})\right|}} \tag{1}$$

We then define the transform matrix $\mathbf{T}_i$ that satisfies:

$$\mathbf{T}_i \mathbf{V}_i = \tilde{\mathbf{V}}_i \tag{2}$$

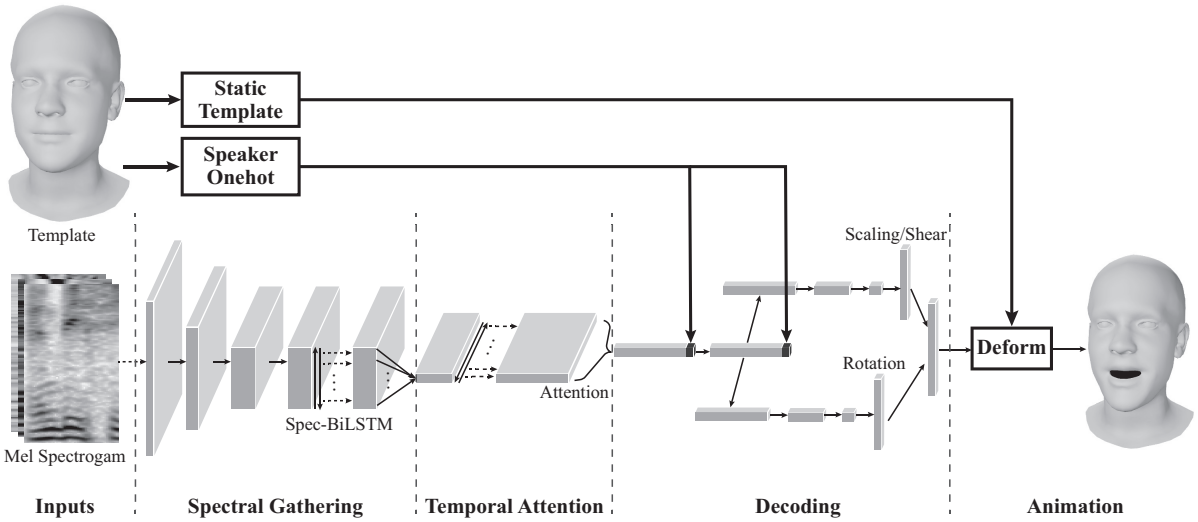


Fig. 1 Our algorithm pipeline.

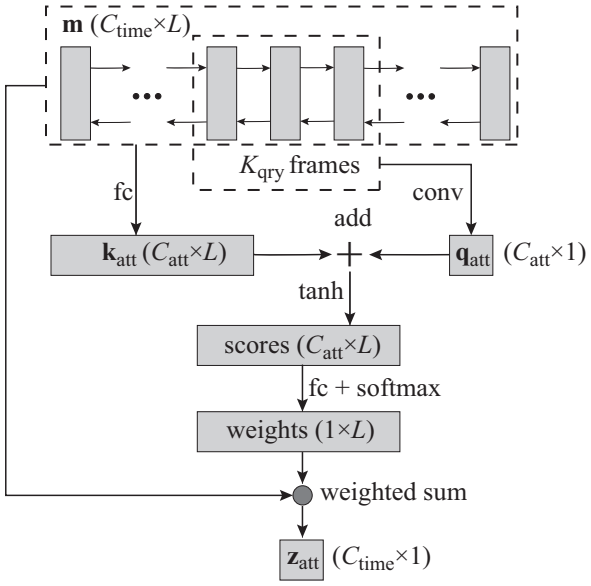


Fig. 2 Temporal attention module. Fully connected layer is represented as 'fc'. 1D convolution layer is represented as 'conv'.

where $\mathbf{V}_i$ and $\tilde{\mathbf{V}}_i$ are three stacked vectors from the template and deformed triangle respectively:

$$\begin{aligned}\mathbf{V}_i &= [v_i^{(2)} - v_i^{(1)} \quad v_i^{(3)} - v_i^{(1)} \quad v_i^{(4)} - v_i^{(1)}] \\ \tilde{\mathbf{V}}_i &= [\tilde{v}_i^{(2)} - \tilde{v}_i^{(1)} \quad \tilde{v}_i^{(3)} - \tilde{v}_i^{(1)} \quad \tilde{v}_i^{(4)} - \tilde{v}_i^{(1)}]\end{aligned} \tag{3}$$

$\mathbf{T}_i$ can be expressed in closed form:

$$\mathbf{T}_i = \tilde{\mathbf{V}}_i \mathbf{V}_i^{-1} \tag{4}$$

Since the rotational component in the transform matrices cannot be directly interpolated, we further perform polar decomposition [19, 20], $\mathbf{T}_i = \mathbf{R}_i\mathbf{S}_i$, to separate $\mathbf{T}_i$ into the *scaling/shear* component $\mathbf{S}_i$ and the *rotational* component $\mathbf{R}_i$. The scaling/shear matrix is symmetric and can be represented with 6 parameters. The rotation, when using Rodrigues' formula, can be expressed in another 3 parameters. Given a template facial mesh with N triangles, the deformation gradients have a total of $9N$ parameters, $6N$-dimensional $\mathbf{s}$ denoting scaling/shear and $3N$-dimensional $\mathbf{r}$ denoting rotation.

To let the model learn from different speakers and also provide the user certain control over the speaking style, we follow VOCA [6] to embed a *speaker-specific label* into the decoding network. Specifically, we first concatenate one-hot coded speaker label to the output of the previous attention module $\mathbf{z}_{att}$ and feed the result to a fully connected layer to get the output $\mathbf{z}_{dec}$. The decoder network then splits into two parallel branches with the same structure to map $\mathbf{z}_{dec}$ to $\mathbf{s}$ and $\mathbf{r}$ separately.

In each branch, $\mathbf{z}_{dec}$ is again concatenated with speaker label and projected by three fully connection layers followed by a final linear layer whose parameters are initialized using the corresponding PCA bases and frozen during training, as shown in **Table 3**.

3.5 Mesh reconstruction

Given the static expressionless template mesh and the deformation gradients $\mathbf{s}$ and $\mathbf{r}$ of a frame, we first convert $\mathbf{s}$ and $\mathbf{r}$ back to per-triangle transform matrices $\{\mathbf{T}_i\}_{i=1}^N$. We then solve the vertex positions $\tilde{\mathbf{x}}$ of the deformed mesh by minimizing

the following energy [18]:

$$E(\mathbf{c}) = \|\mathbf{c} - \mathbf{A}\tilde{\mathbf{x}}\|^2 \tag{5}$$

where $\mathbf{c}$ is the tensor stacked by $\{\mathbf{T}_i\}_{i=1}^N$, and $\mathbf{A}$ is a large, sparse matrix that relates $\tilde{\mathbf{x}}$ to $\mathbf{c}$. $\tilde{\mathbf{x}}$ can be solved in closed form by setting the gradient of $E(\mathbf{c})$ to zero.

$$\mathbf{A}^\mathbf{T}\mathbf{A}\tilde{\mathbf{x}} = \mathbf{A}^\mathbf{T}\mathbf{c} \tag{6}$$

Because $\mathbf{A}$ only depends on the static template, $\mathbf{A}^\mathbf{T}$ and $\mathbf{A}^\mathbf{T}\mathbf{A}$ can be pre-computed once.

4 Model training

4.1 Dataset

We train our model on VOCASET [6], a dataset of 4D face scans with accompanying speech. The dataset totally contains 480 sequences captured at 60 fps from 12 subjects. For each sequence, an aligned mesh sequence is provided, as well as a per-subject static expressionless mesh as template. All meshes share the same topology and contain $N = 9976$ triangles.

We split the data by $8 : 2 : 2$ for training, validation, and test respectively in the same way as original VOCA paper [6]. All sets are fully disjoint, *i.e.*, no overlap of subjects or sentences exists.

For each frame t in a data sequence, we compute the mel spectrogram and the corresponding deformation gradients to form the data pair $(\mathbf{x}_t, \mathbf{y}_t)$, where $\mathbf{y}_t$ can be further split into the scaling/shear term $\mathbf{s}_t \in \mathbb{R}^{6N}$ and the rotation term $\mathbf{r}_t \in \mathbb{R}^{3N}$.

4.2 Loss function

The decoder outputs the deformation gradients as two separate components: $\tilde{\mathbf{s}}_t$ and $\tilde{\mathbf{r}}_t$. For each component, we consider the L2 loss of both the values and the temporal derivatives. The latter encourages temporal smoothness of the results [4, 6].

Specifically, we can define the two loss terms for the scaling/shear component as:

$$L_v^\mathbf{s} = \|\mathbf{s}_t - \tilde{\mathbf{s}}_t\|^2 \tag{7}$$

$$L_d^\mathbf{s} = \|(\mathbf{s}_t - \mathbf{s}_{t-1}) - (\tilde{\mathbf{s}}_t - \tilde{\mathbf{s}}_{t-1})\|^2 \tag{8}$$

The losses for the rotation component, $L_v^\mathbf{r}$ and $L_d^\mathbf{r}$, are defined similarly. The final loss is a weighted sum of the above four terms. The weights are determined automatically using the dynamic scalars proposed by Karras *et al*. [4].

4.3 Data augmentation

We augment our training data in three ways. (i) Similar to Karras *et al*. [4], we randomly shift each frame with ±0.5 frames (about 8.3 ms). For the adjacent frames used to calculate temporal derivative loss terms, we use the same shifting amount to ensure the correctness. (ii) We also follow the common practice in ASR model training to augment the audio signal by randomly adding white or pink noise, and pre-emphasizing the signal with coefficient randomly picked in $[0, 0.95]$. (iii) To cover a wider range of spectral variations, we further apply several augmentation schemes on mel spectrograms: First, we pad zeros in the lowest or the highest frequency bins randomly, then resize into original bins; Second, we randomly squeeze or stretch the time dimension, then resample into the original temporal window; Third, we randomly set some bins to zero; Fourth, scale mel-frequency bins by a random *sin* curve. These augmentations prove to be useful in boosting the robustness of our model.

4.4 Model details

For mel spectrogram extraction, we use $L = 64$ frames and $F = 128$ mel-frequency bins. Each frame processed by short-time Fourier transform has duration of $FFT_{win} = 0.064$ s and is separated by $FFT_{hop} = 0.008$ s. The extracted mel spectrogram represents a window of 0.568 s audio signal, which is enough for capturing coarticulation according to Websdale *et al*. [39] and for handling the audiovisual asynchrony [40].

Table 1 summarizes the layers in formant analysis module. Leaky ReLU activations with leaky rate 0.2 and batch normalization are used for all convolution layers. **Table 2** describes the layers in articulation analysis module. In the motion decoding module, the PCA bases of each branch covers about 97% of the variance, namely 85-dimensional vectors for scaling/shear branch and 180-dimensional vectors for rotation branch. N is 9976 as mentioned in § 4.1. **Table 3** depicts the layers of shared part and two separated branches. Since training set contains 8 subjects, the size of one-hot speaker labels is 8 as well.

For the entire model, weight normalization is performed on all weights.

The model is built with PyTorch [50]. We train the model for 50 epochs using Adam [51] with a constant learning rate of 0.0001 and batch size of 100. In each batch, we randomly choose 50 pairs of adjacent frames to calculate the temporal derivative terms in our loss function. The training takes about 5 hours on a GeForce® GTX 1080 Ti GPU.

Our entire uncompressed model has a memory footprint

of about 66.9 MB. In comparison, the VOCA model [6], with the integrated DeepSpeech module [13], is around 477.3 MB—more than 7 times as large as ours. Among the 66.9 MB memory size, our trainable parameters only hold 26.63 MB and the fixed components from PCA occupy the rest. The small size of trainable parameters can help to avoid overfitting. Furthermore, by converting 32-bit float into 16-bit float and removing some less important PCA components, one can compress the model size greatly and migrate it to mobile devices.

Table 1 Layers of formant analysis module.

Layer	Kernel size[2)]	Stride[2)]	Activation	Output shape
Mel spectrogram	-	-	-	$3 \times 128 \times 64$
Convolution2d	3×1	1×1	lrelu:0.2[1)]	$32 \times 128 \times 64$
MaxPool2d	2×1	2×1	-	$32 \times 64 \times 64$
Convolution2d	3×1	1×1	lrelu:0.2	$64 \times 64 \times 64$
MaxPool2d	2×1	2×1	-	$64 \times 32 \times 64$
Convolution2d	1×1	1×1	lrelu:0.2	$64 \times 32 \times 64$
Spec-BiLSTM	-	-	-	$64 \times 32 \times 64$
Frequency stack	-	-	-	$2048 \times 1 \times 64$
Fully connected	-	-	-	$256 \times 1 \times 64$
Squeezing	-	-	-	256×64

1). lrelu:0.2: Leaky ReLU activation with leaky rate 0.2.
2). Both kernel size and stride are described in *Spectral* × *Temporal* shape.

Table 2 Layers of articulation analysis module.

Layer	Output shape
Time-BiLSTM	512×64
Time-BiLSTM	512×64
Attention	512×1
Squeezing	512

Table 3 Layers of motion decoding module.

Layer	Activation	Output shape (scaling/shear)	Output shape (rotation)
Identity concat	-	520 (shared)	
Fully connected	lrelu:0.2	512 (shared)	
Identity concat	-	520	520
Fully connected	lrelu:0.2	512	512
Fully connected	tanh	256	256
Fully connected	-	85	180
Inverse PCA	-	59865	29928

5 Evaluation

Quantitative assessment of a speech-driven facial animation is difficult for several reasons. First, the mapping between speech signals and facial motion is known to be ambiguous [6] and affected greatly by the speaker. Second, humans have developed outstanding capability of detecting nuanced facial expressions [52]. A trivial error in the *geometrical* sense may be picked up immediately by human eyes as something uncannily wrong [53]. This may explain why *none* of most relevant works (*e.g.*, by Karras *et al*. [4] and Cudeiro *et al*. [6]) includes a quantitative evaluation. Therefore, we conduct several user studies and qualitative evaluation.

5.1 User studies

Three blind user studies are published on Amazon Mechanical Turk (AMT) in the form of A/B choices. In each study a user is presented with two side-by-side synchronized animation clips driven by the same audio, and asked to choose the better one. The template meshes and rendering configurations remain the same for both results. The user can also play the animation at half speed to better compare motion details.

To prevent some participants from randomly picking answers without careful evaluation, we adopt two measures. First, a user can only give answer after watching both clips. Second, several pairs with an obvious answer are included as qualification questions, and we only accept answers from users who have passed the qualification.

We have collected a total of 5600 HITs (human intelligence task), each represents one participant making one choice between two clips. Notably, the number of both utterances and collected HITs in our user studies are times larger than similar ones conducted by Cudeiro *et al*. [6].

5.1.1 Comparison with captured data

In the first study we want to know how close animations generated from speech using our method and the "ground truth" are. We utilize the test sequences in VOCASET [6] and ask the participants to choose between the captured facial animation and those produced by our method conditioned on all eight speaker styles. Among 800 HITs collected, users prefer the captured results (81.25% ± 3.78%) over ours—a number quite similar to that reported in prior art [6]. This is not unexpected though, as certain visual subtleties are known to be absent in the speech audio [3].

5.1.2 Deformation gradients vs. vertex offsets

In our own comparison we have noticed that using deformation gradients as the network output works equally well as vertex offsets do most of the time. But in certain cases, deformation gradients noticeably outperforms the other representation. To confirm this observation, we conduct a second user study where participants are asked to compare 48 pairs of results (12 sentences in 4 random styles) generated using

these two motion representations, with audio sources from previous work and YouTube.

We have collected 50 HITs for each result pair, totaling 2400 HITs. Overall, the participants' preference lean only slightly toward deformation gradients (53.58%±3.42%). But interestingly, deformation gradients seem to work significantly better in sentences involving transition from the phoneme /s/ to any of /m/, /b/, or /p/. As shown in **Fig. 3**, the first five columns from the left are sentences with the aforementioned phonetic transitions. Please watch the supplementary video for dynamic comparison.

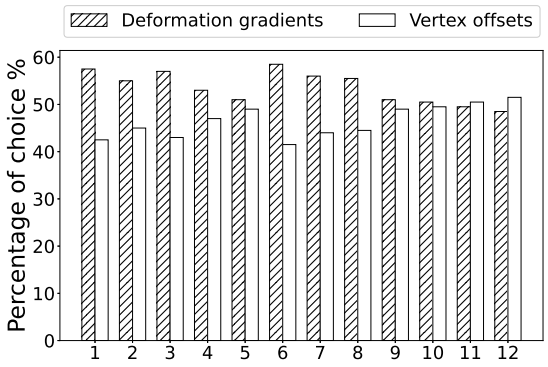


Fig. 3 User study comparing *deformation gradients* and *vertex offsets*.

5.1.3 Comparison with VOCA

In the third study we compare our method with VOCA [6], with a focus on the two methods' robustness regarding audio quality. We randomly select 12 sentences (different from those in the second study), each in 4 random styles. From the test set, we sampled four sentences those have similar recording quality to the training data. From the supplementary videos of previous works, we select two clips with no noise, one clip with medium noise (-24dB), and one clip with slightly higher (-18dB) noise. Besides, four clips from YouTube are considered. Two of them have low background music and the other two are recorded in open space. The clips of VOCA are generated with their code and released model. In total, the 2400 HITs show no obvious preference between VOCA (45.14% ± 10.02%) and our method (54.86% ± 10.02%). But as **Fig. 4** illustrates, VOCA handles high noise level and open space environment much better than our method does, probably due to the exhaustive speech data used to train their integrated DeepSpeech module. Our lightweight model, on the other hand, works surprisingly well when the noise level is relatively low, and handles different languages, genders, pitches etc. robustly.

Furthermore, our proposed model outperforms VOCA [6] at several special cases where wordless singing or long-lasting vowel exists, as shown in our supplementary video. The failure of VOCA at these cases is probably caused by the failure of ASR module, DeepSpeech, because wordless singing and long-lasting vowel are quite hard for ASR tasks.

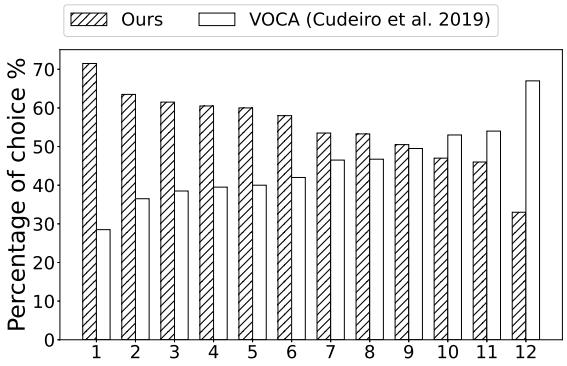


Fig. 4 User study comparing our method with VOCA [6]. The columns are sorted roughly by increasing noise level in the audio.

5.2 Qualitative evaluation

Please watch the video in our supplementary materials for all the dynamic results mentioned in this section. None of the results contains utterances from the training set.

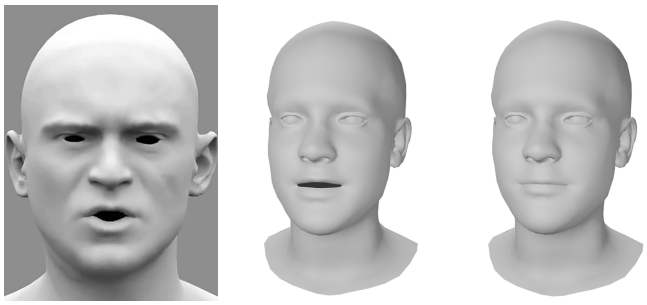
5.2.1 Comparison with Karras *et al.*

We also compare our results with Karras *et al.*'s [4]. Since their model and data are not publicly available yet, we have both compared directly to the clips in their supplementary video, and implemented their algorithm ourselves, but having the original emotional states replaced by the one-hot coded speaker labels and training with the same VOCASET which our own model consumes.

As shown in **Fig. 5** and the supplementary video, our model handles plosives particularly better in comparison with theirs.

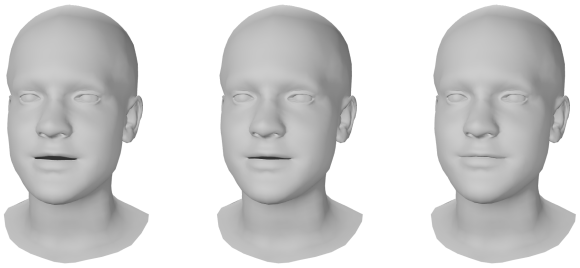
5.2.2 Ablation study

We perform ablation study on the key modifications we have made in the network architecture. The Spec-BiLSTM and temporal attention are replaced by convolutional layers along spectral and temporal dimensions respectively, and in both cases significant downgrades of the result quality can be observed. If Spec-BiLSTM is replaced by convolution layers,



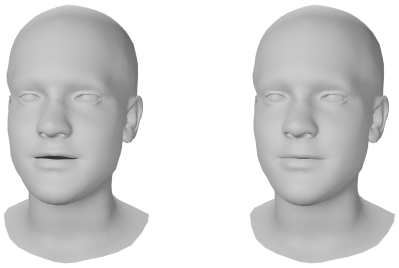
(a) Karras *et al*. (b) Their model+our data (c) Ours

Fig. 5 Comparison with Karras *et al*. [4]. The speaker is about to pronounce the phoneme /b/ at this moment, when the lips are supposedly pressed.



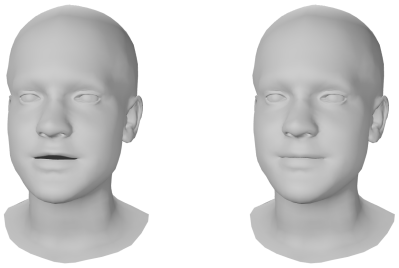
(a) No Spec-BiLSTM (b) No temporal attention (c) Our full model

Fig. 6 Ablation of model architecture. Consonant /p/ from a test sentence is pronounced at this moment.



(a) Without audio augment (b) Our full model

Fig. 7 Data augmentation in audio signal. Consonant /b/ from a test sentence is pronounced at this moment.



(a) Without mel augment (b) Our full model

Fig. 8 Data augmentation in mel spectrograms. Consonant /p/ from a test sentence is pronounced at this moment.

as shown in **Fig. 6(a)**, the lips fail to press at the consonant /p/. **Fig. 6(b)** depicts a similar case when temporal attention is replaced by convolution layers.

We also compare models trained without and with data augmentation. Without data augmentation, the model is more likely to fail at certain consonants, especially plosives. **Fig. 7** shows when /b/ is pronounced, the model without audio signal augmentation fails to press lips. **Fig. 8** shows another case where /p/ is pronounced, the model without mel spectrogram augmentation fails to press lips as well.

More results of ablation study can be found in our supplementary video, which conveys the comparison in a dynamic and noticeable way.

5.2.3 Attention analysis

Fig. 9 shows visualizations of internal weights in our temporal attention module when processing two distinct phonemes. It is interesting to see that when processing a vowel, the weights are almost evenly distributed (*e.g*., /ei/ in **Fig. 9(a)**). However, when the input window is occupied by a consonant, the center frames that correspond to the *plosive* attract exclusive attention (*e.g*., /b/ in **Fig. 9(b)**)—a phenomenon that matches one's intuition well.

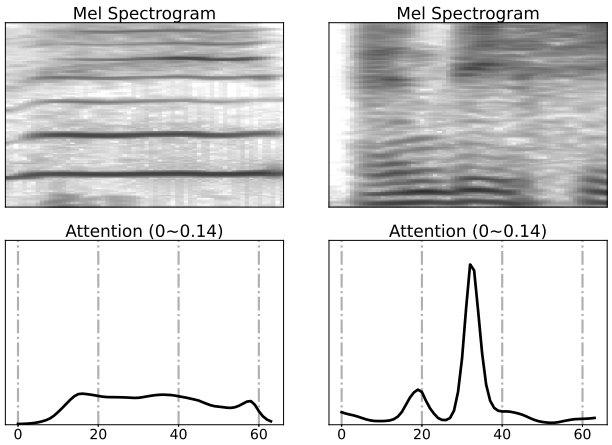
5.2.4 Topology-independent retargeting

We have also applied the output facial motion, represented in deformation gradients, to new 3D avatars that have drastically different mesh topologies by following the method mentioned in [18]. The results can be seen in **Fig. 10** and the supplementary video.

5.3 Running time

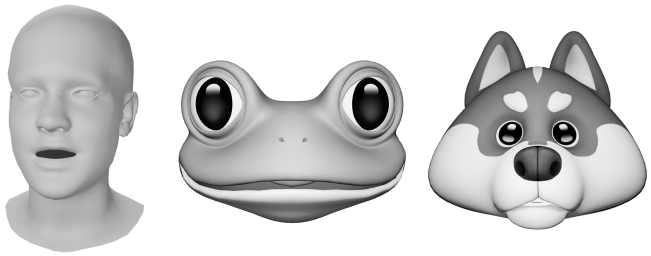
At runtime, we compare the time consumptions of different approaches (Karras *et al*.'s [4], VOCA [6] and ours) to generate an animation from the same input audio (5,095 ms). As shown in **Table 4**, we test all the three models on CPU because the DeepSpeech module used by VOCA and all the preprocessing steps run on CPU. The neural networks complete inference of all frames in a single batch.

In addition to the total running time, we also measure the time comsumptions for individual algorithm stages. Our method uses deformation gradients as motion representation, and thus requires two extra stages to set the static mesh tem-



(a) Vowel '/ei/' at center (b) Consonant '/b/' at center

Fig. 9 Visualization of attention module at vowel and consonant.



(a) Original mesh (b) Frog (c) Dog

Fig. 10 The generated facial motions applied to other artist designed meshes with different topologies.

plate and reconstruct the final mesh from predicted deformation gradients. During preprocessing Karras *et al*.'s [4], VOCA [6] and ours compute LPC, MFCCs and mel spectrograms, respectively. To extract audio features, VOCA uses DeepSpeech, while Karras *et al*.'s and ours use custom audio encoders. Overall, the running time of VOCA is about twice as that of our method mainly due to the overhead of DeepSpeech. Karras *et al*.'s [4] remains the fastest because of its fully convolutional structure.

On a GeForce® GTX 1080 Ti GPU, our method takes about 10 ms to generate one complete facial mesh frame, including mel spectrogram computation, network inference (with a batch size of 1), and mesh reconstruction. Even though the current implementation is able to perform real-time speech-driven animation, there is still much room for optimization.

Table 4 Comparison of time consumptions (ms).

Stage	Karras *et al*.'s [4]	VOCA [6]	Ours
Set static mesh	-	-	13.15
Preprocess audio	1,672.15	8.94	553.92
Get audio feature	376.09	9,253.29	4,200.69
Get anime feature	51.06	578.35	545.03
Reconstruct mesh	-	-	589.57
Total (with loading)	2,531.80	11,999.58	6,425.50

The same input audio (5,095 ms) is used. All tests run on an Intel® Core™ i7-8700K (@3.70 GHz) CPU. Karras *et al*.'s [4] is implemented by us due to the lack of official release. The original authors' code and pre-trained model of VOCA [6] are used for the test.

6 Limitation and future work

In order to reach a higher level of realism, there are several other aspects that this paper does not account for. For example, the abilities to easily control the emotion or speaking style would greatly enhance the expressiveness of the facial animation. Some recent studies [54–59] exploit the association between faces and voices, aiming to identify a person or detect the emotional state from face images/videos and voice in a synergetic fashion. Oh *et al*. [57] attempt to infer the speaker's face from the input voice. In the future, it may be possible to automatically infer a person-specific and expressionless facial mesh, and animate it with given voice while allowing speaking style and emotional state to be further controlled.

The motions our method learns and infers focus primarily in the *lip and jaw* area. But evidences show that there are also correlations (albeit weaker) between speech and other subtle motions [3], including eye gaze, head motion, gestures [60] etc. It is interesting to see how much of these spontaneous and non-vocal conversational signals a model can learn and reproduce.

7 Conclusion

We present a novel speech-driven facial animation algorithm. A spectral-dimensional bidirectional long short-term memory is introduced to exploit formant features spanning a wide spectrogram. Frame-wise attention mechanism is used to better synchronize lip motions with vocal signals in the temporal dimension. Together they form an efficient and lightweight vocal feature encoder suitable for our specific task. We use deformation gradients to represent facial motion. Results show that it handles highly non-rigid deformations, like those

around pressed lips, especially well. Our model is significantly smaller than those featuring pre-trained ASR modules, but offers comparable robustness and higher quality results in certain challenging cases.

Acknowledgements We would like to thank VOCA group for publishing their database. This work is partially supported by the National Key Research & Development Program of China (2016YFB1001403) and NSF China (No. 61572429).

References

1. Chen Cao, Qiming Hou, and Kun Zhou. Displaced dynamic expression regression for real-time facial tracking and animation. *ACM Transactions on graphics (TOG)*, 33(4), July 2014.
2. Koki Nagano, Shunsuke Saito, Lain Goldwhite, Kyle San, Aaron Hong, Liwen Hu, Lingyu Wei, Jun Xing, Qingguo Xu, Han-Wei Kung, Jiale Kuang, Aviral Agarwal, Erik Castellanos, Jaewoo Seo, Jens Fursund, and Hao Li. Pinscreen avatars in your pocket: Mobile pagan engine and personalized gaming. In *SIGGRAPH Asia 2018 Real-Time Live!*, SA '18, New York, NY, USA, 2018. Association for Computing Machinery.
3. Pif Edwards, Chris Landreth, Eugene Fiume, and Karan Singh. Jali: An animator-centric viseme model for expressive lip synchronization. *ACM Transactions on graphics (TOG)*, 35(4), July 2016.
4. Tero Karras, Timo Aila, Samuli Laine, Antti Herva, and Jaakko Lehtinen. Audio-driven facial animation by joint end-to-end learning of pose and emotion. *ACM Transactions on Graphics (TOG)*, 36(4), July 2017.
5. Hai Xuan Pham, Yuting Wang, and Vladimir Pavlovic. End-to-end learning for 3d facial animation from speech. In *Proceedings of the 20th ACM International Conference on Multimodal Interaction*, ICMI '18, pages 361–365, New York, NY, USA, 2018. Association for Computing Machinery.
6. Daniel Cudeiro, Timo Bolkart, Cassidy Laidlaw, Anurag Ranjan, and Michael J. Black. Capture, learning, and synthesis of 3d speaking styles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10101–10111, June 2019.
7. Yliess Hati, Francis Rousseaux, and Clément Duhart. Text-driven mouth animation for human computer interaction with personal assistant. In *ICAD 2019: The 25th International Conference on Auditory Display*, pages 75–82. Department of Computer and Information Sciences, Northumbria University, 2019.
8. Dan Jurafsky and James H. Martin. *Speech and language processing : an introduction to natural language processing, computational linguistics, and speech recognition*. Pearson Prentice Hall, Upper Saddle River, N.J., 2009.
9. Supasorn Suwajanakorn, Steven M. Seitz, and Ira Kemelmacher-Shlizerman. Synthesizing obama: Learning lip sync from audio. *ACM Transactions on Graphics (TOG)*, 36(4), July 2017.
10. Sarah Taylor, Taehwan Kim, Yisong Yue, Moshe Mahler, James Krahe, Anastasio Garcia Rodriguez, Jessica Hodgins, and Iain Matthews. A deep learning approach for generalized speech animation. *ACM Transactions on Graphics (TOG)*, 36(4), July 2017.
11. Ahmed Hussen Abdelaziz, Barry-John Theobald, Justin Binder, Gabriele Fanelli, Paul Dixon, Nick Apostoloff, Thibaut Weise, and Sachin Kajareker. Speaker-independent speech-driven visual speech synthesis using domain-adapted acoustic models. In *2019 International Conference on Multimodal Interaction*, ICMI '19, pages 220—-225, New York, NY, USA, 2019. Association for Computing Machinery.
12. Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
13. Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*, 2014.
14. Hai X Pham, Samuel Cheung, and Vladimir Pavlovic. Speech-driven 3d facial animation with implicit emotional awareness: a deep learning approach. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2328–2336. IEEE, 2017.
15. Guanzhong Tian, Yi Yuan, and Yong Liu. Audio2face: Generating speech/face animation from single audio with attention-based bidirectional lstm networks. In *2019 IEEE international conference on Multimedia & Expo Workshops (ICMEW)*, pages 366–371. IEEE, 2019.
16. Panagiotis Tzirakis, Athanasios Papaioannou, Alexander Lattas, Michail Tarasiou, Björn Schuller, and Stefanos Zafeiriou. Synthesising 3d facial motion from" in-the-wild" speech. *arXiv preprint arXiv:1904.07002*, 2019.
17. Ryosuke Nishimura, Nobuchika Sakata, Tomu Tominaga, Yoshinori Hijikata, Kensuke Harada, and Kiyoshi Kiyokawa. Speech-driven facial animation by lstm-rnn for communication use. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 1102–1103. IEEE, 2019.
18. Robert W Sumner and Jovan Popović. Deformation transfer for triangle meshes. *ACM Transactions on graphics (TOG)*, 23(3):399–405, 2004.
19. Qianyi Wu, Juyong Zhang, Yu-Kun Lai, Jianmin Zheng, and Jianfei Cai. Alive caricature from 2d to 3d. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7336–7345, 2018.
20. Lin Gao, Y. Lai, Jie Yang, Ling-Xiao Zhang, L. Kobbelt, and S. Xia. Sparse data driven mesh deformation. *IEEE transactions on visualization and computer graphics*, 2019.
21. Verónica Orvalho, Pedro Bastos, Frederic I Parke, Bruno Oliveira, and Xenxo Alvarez. A facial rigging survey. In *Eurographics (STARs)*, pages 183–204, 2012.
22. Raymond D Kent and Fred D Minifie. Coarticulation in recent speech production models. *Journal of phonetics*, 5(2):115–133, 1977.
23. Catherine Pelachaud, Norman I Badler, and Mark Steedman. Generating facial expressions for speech. *Cognitive science*, 20(1):1–46, 1996.
24. Alice Wang, Michael Emmi, and Petros Faloutsos. Assembling an

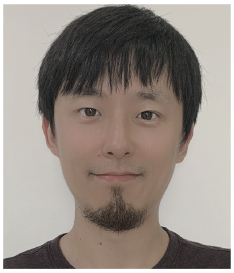
expressive facial animation system. In *Proceedings of the 2007 ACM SIGGRAPH symposium on Video games*, pages 21–26. ACM, 2007.

25. Michael M Cohen and Dominic W Massaro. Modeling coarticulation in synthetic visual speech. In *Models and techniques in computer animation*, pages 139–156. Springer, 1993.
26. Yuyu Xu, Andrew W Feng, Stacy Marsella, and Ari Shapiro. A practical and configurable lip sync method for games. In *Proceedings of Motion on Games*, pages 131–140. ACM, 2013.
27. Christoph Bregler, Michele Covell, and Malcolm Slaney. Video rewrite: Driving visual speech with audio. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 353–360, 1997.
28. Tony Ezzat, Gadi Geiger, and Tomaso Poggio. Trainable videorealistic speech animation. *ACM Transactions on Graphics (TOG)*, 21(3):388–398, July 2002.
29. Sarah L Taylor, Moshe Mahler, Barry-John Theobald, and Iain Matthews. Dynamic units of visual speech. In *Proceedings of the 11th ACM SIGGRAPH/Eurographics conference on Computer Animation*, pages 275–284, 2012.
30. Matthew Brand. Voice puppetry. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 21–28. ACM Press/Addison-Wesley Publishing Co., 1999.
31. Lei Xie and Zhi-Qiang Liu. Realistic mouth-synching for speech-driven talking face using articulatory modelling. *IEEE Transactions on Multimedia*, 9(3):500–510, 2007.
32. Lijuan Wang, Wei Han, Frank K Soong, and Qiang Huo. Text driven 3d photo-realistic talking head. In *Interspeech 2011*, pages 3307–3308, 2011.
33. Xinjian Zhang, Lijuan Wang, Gang Li, Frank Seide, and Frank K Soong. A new language independent, photo-realistic talking head driven by voice only. In *Interspeech 2013*, pages 2743–2747, 2013.
34. Taiki Shimba, Ryuhei Sakurai, Hirotake Yamazoe, and Joo-Ho Lee. Talking heads synthesis from audio with deep neural networks. In *2015 IEEE/SICE International Symposium on System Integration (SII)*, pages 100–105. IEEE, 2015.
35. Bo Fan, Lei Xie, Shan Yang, Lijuan Wang, and Frank K Soong. A deep bidirectional lstm approach for video-realistic talking head. *Multimedia Tools and Applications*, 75(9):5287–5309, 2016.
36. Sefik Emre Eskimez, Ross K Maddox, Chenliang Xu, and Zhiyao Duan. Generating talking face landmarks from speech. In *International Conference on Latent Variable Analysis and Signal Separation*, pages 372–381. Springer, 2018.
37. Deepali Aneja and Wilmot Li. Real-time lip sync for live 2d animation. *arXiv preprint arXiv:1910.08685*, 2019.
38. David Greenwood, Iain Matthews, and Stephen Laycock. Joint learning of facial expression and head pose from speech. In *Proc. Interspeech 2018*, pages 2484–2488, 2018.
39. Danny Websdale, Sarah Taylor, and Ben Milner. The effect of real-time constraints on automatic speech animation. In *Proc. Interspeech 2018*, pages 2479–2483, 2018.
40. Jean-Luc Schwartz and Christophe Savariaux. No, there is no 150 ms lead of visual speech on auditory speech, but a range of audiovisual asynchronies varying from small audio lead to large audio lag. *PLOS Computational Biology*, 10(7):e1003743, 2014.
41. Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4779–4783. IEEE, 2018.
42. Ryan Prenger, Rafael Valle, and Bryan Catanzaro. Waveglow: A flow-based generative network for speech synthesis. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3617–3621. IEEE, 2019.
43. Konstantinos Vougioukas, Stavros Petridis, and Maja Pantic. Realistic speech-driven facial animation with gans. *International Journal of Computer Vision*, 128(5):1398–1413, May 2020.
44. Lele Chen, Ross K Maddox, Zhiyao Duan, and Chenliang Xu. Hierarchical cross-modal talking face generation with dynamic pixel-wise loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7832–7841, 2019.
45. Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, and Dong Yu. Convolutional neural networks for speech recognition. *IEEE/ACM Transactions on audio, speech, and language processing*, 22(10):1533–1545, 2014.
46. Tara N. Sainath and Bo Li. Modeling time-frequency patterns with lstm vs. convolutional architectures for lvcsr tasks. In *Interspeech 2016*, pages 813–817, 2016.
47. Yuzhou Liu and DeLiang Wang. Time and frequency domain long short-term memory for noise robust pitch tracking. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5600–5604. IEEE, 2017.
48. Misha Denil, Loris Bazzani, Hugo Larochelle, and Nando de Freitas. Learning where to attend with deep architectures for image tracking. *Neural computation*, 24(8):2151–2184, 2012.
49. Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.
50. Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS 2017 Workshop on Autodiff*, 2017.
51. Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
52. Paul Ekman, Wallace V Friesen, and Joseph C Hager. *Facial action coding system: The manual on CD-ROM. Instructor's Guide.* Salt Lake City: Network Information Research Co., 2002.
53. Masahiro Mori, Karl F MacDorman, and Norri Kageki. The uncanny valley [from the field]. *IEEE Robotics & Automation Magazine*, 19(2):98–100, 2012.
54. Changil Kim, Hijung Valentina Shin, Tae-Hyun Oh, Alexandre Kaspar, Mohamed Elgharib, and Wojciech Matusik. On learning associations of faces and voices. In *Proceedings of Asian Conference on Computer Vision (ACCV)*, pages 276–292. Springer, 2018.

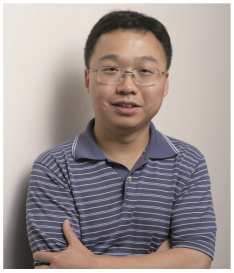
55. Valentin Vielzeuf, Corentin Kervadec, Stéphane Pateux, Alexis Lechervy, and Frédéric Jurie. An occam's razor view on learning audiovisual emotion recognition with small training sets. In *Proceedings of the 20th ACM International Conference on Multimodal Interaction*, pages 589–593, 2018.
56. Egils Avots, Tomasz Sapiński, Maie Bachmann, and Dorota Kamińska. Audiovisual emotion recognition in wild. *Machine Vision and Applications*, 30(5):975–985, 2019.
57. Tae-Hyun Oh, Tali Dekel, Changil Kim, Inbar Mosseri, William T Freeman, Michael Rubinstein, and Wojciech Matusik. Speech2face: Learning the face behind a voice. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7539–7548, 2019.
58. Rui Wang, Xin Liu, Yiu-ming Cheung, Kai Cheng, Nannan Wang, and Wentao Fan. Learning discriminative joint embeddings for efficient face and voice association. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, pages 1881–1884, New York, NY, USA, 2020. Association for Computing Machinery.
59. Hao Zhu, Mandi Luo, Rui Wang, Aihua Zheng, and Ran He. Deep audio-visual learning: A survey. *arXiv preprint arXiv:2001.04758*, 2020.
60. Shiry Ginosar, Amir Bar, Gefen Kohavi, Caroline Chan, Andrew Owens, and Jitendra Malik. Learning individual styles of conversational gesture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3497–3506, 2019.

Yujin Chai received his B.S. degree in the Computer Science and Technology Department from Zhejiang University, China in 2016. He is now a Ph.D. student at the State Key Lab of CAD&CG, Zhejiang University. His research interests include machine learning and facial animation.

Yanlin Weng is currently an Associate Professor of the School of Computer Science and Technology at Zhejiang University. She got her Ph.D. degree in Computer Science from University of Wisconsin - Milwaukee, and her master and bachelor degrees in Control Science and Engineering from Zhejiang University. Her research interest includes computer graphics and multimedia.



Lvdi Wang is currently the Director of FaceUnity Research Center of Beijing, China. He received his Ph.D. degree in Computer Science from Institute for Advanced Study, Tsinghua University in 2011, under the supervision of Prof. Baining Guo. After that, he has spent four years at Microsoft Research Asia and two years at Apple Inc. His works focus on computer graphics, speech, and natural language processing techniques.



Kun Zhou is a Cheung Kong Professor in the Computer Science Department of Zhejiang University, and the Director of the State Key Lab of CAD&CG. Prior to joining Zhejiang University in 2008, Dr. Zhou was a Leader Researcher of the Internet Graphics Group at Microsoft Research Asia. He received his B.S. degree and Ph.D. degree in computer science from Zhejiang University in 1997 and 2002, respectively. His research interests are in visual computing, parallel computing, human computer interaction, and virtual reality. He is a Fellow of IEEE.