Speaker-independent Speech Animation using Perceptual Loss Functions and Synthetic Data

Danny Websdale, Sarah Taylor and Ben Milner

***Abstract*—We propose a real-time speaker-independent speech-to-facial animation system that predicts lip and jaw movements on a reference face for audio speech taken from any speaker. Our approach is motivated by two key observations; 1) Speaker-independent facial animation can be generated from phoneme labels, but to perform this automatically a speech recogniser is needed which, due to contextual look-ahead, introduces too much time lag. 2) Audio-driven speech animation can be performed in real-time but requires large, multi-speaker audio-visual speech datasets of which there are few. We adopt a novel three-stage training procedure that leverages the advantages of each approach. First we train a *phoneme*-to-visual speech model from a large single-speaker audio-visual dataset. Next, we use this model to generate the synthetic visual component of a large multi-speaker audio dataset of which the video is not available. Finally, we learn an *audio*-to-visual speech mapping using the synthetic visual features as the target. Furthermore, we increase the realism of the predicted facial animation by introducing two perceptually-based loss functions that aim to improve mouth closures and openings. The proposed method and loss functions are evaluated objectively using mean square error, global variance and a new metric that measures the extent of mouth opening. Subjective tests are carried out over the best performing systems. Results show that our approach produces audio-driven facial animation that is comparable to those produced from phoneme sequences and that improved mouth closures, particularly for bilabial closures, are achieved.**

Index terms - speech-to-facial animation, speaker-independent, BLSTM, perceptual loss functions, avatars, talking heads, speech animation.

I. Introduction

Speech animation is the process of creating facial motion on a digital character that synchronises to given audio speech. It is an essential component of animated television shows, movies and computer games, and is traditionally created either manually by artists or semi-automatically using motion capture technology. In recent years, a range of automated methods has been proposed that allow facial animation to be estimated directly from the audio speech signal [1], [2], [3], [4], or from a text or phoneme sequence [5], [6]. Automatically generated facial animation can be produced much faster than by hand and has the potential for real-time processing which makes it well suited to animating virtual characters online in a range of interactive multi-modal human-machine interfaces such as computer games, intelligent assistants, and virtual and augmented reality [7].

The challenge of automatic speech animation has been around for decades. Early attempts were rule-based systems such as phoneme-driven key-frame interpolation [8], [9], [10] or sample concatenation [11], [12], [13]. A notable example was *Video Rewrite* which was a video-based solution based on blending video samples of the mouth region corresponding to triphones and stitching it onto a full-face video background [11]. Rule-based approaches tended to lack realism since animation is constrained to a set of predefined poses. More compelling facial animation has since been achieved using data-driven model-based systems [6], [14], [15], [16] and we base our work on this approach.

Automatic speech animation can be driven from text or from an audio signal. With animation from text, an intermediate linguistic representation (typically phoneme labels) is first created. A model is then trained to estimate sequences of visual speech parameters from the phoneme labels [16]. With animation from an audio signal, one method is to extract a phonetic transcription using an automatic speech recogniser [10] and then estimate visual parameters. Alternatively, automatic speech animation can be achieved by learning a mapping directly from the audio waveform [2], [17], [18] or from acoustic features (e.g filterbank) [19], [20] to the visual parameters. An advantage of using a linguistic representation, such as phonemes, is that they are inherently speaker independent since they do not encode the speaker's identity, only the speech content. A disadvantage is that a robust speech recogniser is needed to automatically produce an accurate phoneme transcription. If the speech recogniser is trained across a range of speakers, then speaker-independent animation is possible, although the phoneme accuracy is likely to be lower than a speaker-dependent system. However, large vocabulary, robust speech recognisers require broad contextual windows that span multiple words in order to exploit language models which improve decoding accuracy [21]. This introduces a significant time lag that makes real-time speech animation prohibitive, where tolerable delays are of the order 200ms [22], [23]. Conversely, audio-driven speech animation has been shown to generate realistic lip synchronisation in real-time, but in a speaker-dependent setting [2], [23]. Our work leverages the advantages of both audio and text driven methods to achieve speaker-independent, real-time speech animation.

To animate speech from any speaker in real-time, essentially two options can be considered. One approach is to employ a set of person-specific speaker-dependent models. This may be expected to generate good animations for a given speaker but would be challenging to implement. Sufficient training data for each person-specific model would be required and this would need to be added to for any new speakers. Furthermore, input speech would need to be classified in terms

School of Computing Sciences, University of East Anglia, UK e-mail: {d.websdale, s.l.taylor, b.milner}@uea.ac.uk

of the speaker before being sent to the appropriate model. Instead, our approach is to create a single speaker-independent model that maps audio from any speaker directly into visual parameters.

The first aim of this work is to develop a framework that maps audio speech, taken from any speaker, to a set of visual parameters corresponding to the pose of a representative (target) face. We harness the speaker-independent quality of phoneme-to-animation systems to generate synthetic training data with which to train our audio-driven system. All speakers are animated to a fixed representative face so that a graphics character need only be set up once and can be controlled by a voice from any speaker. This is beneficial for applications such as computer games, where an avatar is rigged once and can subsequently be controlled by any player's voice. An alternative approach would be to use a speaker-independent speech recogniser to generate a sequence of phonemes, and their start and end times, and input this into a phoneme-to-visual parameter mapping. However, the broad contextual windows required in such a system would introduce substantial delay compared to our proposed method. Further, any phoneme errors in the output transcription would lead to erroneous visual parameters as would errors in the phoneme start and end times. No such 'hard' error exists in the proposed approach as the mapping does not go through an intermediate symbol (phoneme) mapping and instead maps directly from audio features to visual parameters.

The optimisation algorithms in model-based systems typically minimise the mean squared error [6] or maximise the correlation [24] between predicted and target values. Existing loss functions are computed over the entire face (or jaw) region. However, our initial tests found that poor facial animation is particularly evident for bilabial phonemes (/b/, /p/, /m/) which require a full mouth closure to appear realistic. Therefore, the second aim of our work is to improve the realism of the predicted animation by developing loss functions that put emphasis on perceptually significant events such as bilabial closures. The main contributions of this paper can be summarised as:

- We introduce a modular framework for generating speaker-independent speech animation.
- We exploit phoneme-driven speech animation to generate synthetic training data.
- We develop a robust speaker-independent audio-driven speech animation system that is trained on synthetic data.
- We introduce perceptually motivated loss functions to increase the naturalness of predicted facial animation.
- We introduce a new evaluation metric to measure explicitly the extent of mouth opening.

The remainder of this paper is organised as follows. Related work is discussed in Section II. Section III explains the methods of acoustic and visual feature extraction. The proposed speaker-independent audio-to-visual mapping is introduced in Section IV. Section V describes the two perceptually motivated loss functions to improve the naturalness of predicted animation. Experimental results are presented in Section VI where we use both objective and subjective analysis to evaluate speaker-dependent and speaker-independent animation. Finally, a discussion is made of the proposed methods in Section VII and the work concluded in Section VIII.

II. Related work

Some of the earliest model-based methods for estimating visual parameters from audio used hidden Markov models (HMMs) [3], [25], [26], [27], [28]. However, these methods were limited by their use of Gaussian mixture model-based states (with the assumption of diagonal covariances) and by the decision tree clustering of visual features within states [29], and consequently produced animation that lacked expressiveness. More recently, deep neural networks (DNNs) have been shown to be effective at learning the non-linear mappings between input speech audio or linguistic features, and output visual parameters [2], [6], [30], [31]. A facial pose depends not only on the current sound, but also upon the neighbouring sounds due to co-articulation, so networks are typically trained to learn a mapping from a *window* of input speech to the visual parameters.

Alternatively, recurrent neural networks (RNNs) inherently capture neighbouring sounds using an internal state that remembers past events. Furthermore, bi-directional RNNs (BRNNs) can use both past and future inputs to make a prediction, and have been applied successfully to audio speech synthesis [32]. A well-known issue with RNNs is that propagated gradients can become very small and vanish, particularly when modelling long span relationships. The long short-term memory (LSTM) model overcomes this by using a series of gates to control the flow of information [33]. LSTMs (and bi-directional LSTMs) have been applied successfully to automatic speech animation [34], [24], [18] as well as audio-driven head motion synthesis [18], [35], [36]. A three-stage LSTM has been used to predict the parameters required for the JALI 3D facial animation rig [37], [38]. One LSTM is used map the input audio into phoneme groups while a second LSTM maps the audio into facial landmarks. Finally a third LSTM combines these two outputs with the audio to generate the JALI parameters. Convolution neural networks (CNNs) have also been applied to facial animation. An architecture comprising first a series of CNNs, with 1D filter kernels, and a fully connected layer has been used to map raw audio into active shape model (ASM) parameters [39]. To give smoother changes between frames, the previous ASM parameter vector is fed back and used in the prediction of the current frame. Alternatively, in [40], a series of CNNs are used to map input spectrograms into a feature representation which is then input into an LSTM and dense layers to predict facial parameters.

Other model-based approaches include generative adversarial networks (GANs) in which one or more discriminators are trained to detect fake animation and a generator is simultaneously trained to produce animation good enough to fool the detector. Vougioukas et al. [41] and Sadoughi and Busso [42] applied this technique to video-based animation and expressive lip animation respectively. A cascaded GAN approach has also been proposed whereby input audio is first mapped to facial landmark features, which are then mapped to the final video [43]. This is shown to synthesise more robust animation that has fewer visual artefacts that arise from irrelevant sounds in the audio. Variational Autoencoders (VAEs) were used by [44] to generate speech animation from both audio and gaze. A multimodal architecture was used to embed all input modalities and the facial coefficients in the shared latent space.

III. Audio and visual feature extraction

Automatic speech animation is a transformation from audio speech to facial motion. In this work we pose this as the task of mapping from a set of acoustic speech *features* to visual speech *features*. To achieve realistic animation the visual features must adequately encode the speaker's facial pose while the acoustic features must contain sufficient speech-related information to regress this pose.

A. Visual features

Speech-to-animation models can be trained to predict facial animation in various forms. Image-based systems output pixel values in an effort to create video realistic output [17]. Others output directly to the vertices of a 3D face mesh [2]. More commonly, models are trained to predict a parameterisation of a face that describes the geometry or appearance of the pose [6]. We use a parametric representation of the face that is derived from an active appearance model (AAM) [45]. As a means of representing faces, AAMs have had widespread success [6], [45], [46], [47]. AAMs are particularly suited to our approach because they separate facial actions in the feature space (for example the first dimension of the AAM representation describes the extent of lip opening and closing - see Figure 3) which facilitates the calculation of our novel perceptual loss, defined in Section V. Another benefit of AAMs is that the predicted animation can be easily retargeted to a graphics character by transferring the AAM to the character as a blendshape rig [6].

The specific implementation of the AAM used in our work is described in [12]. In summary, the AAM is trained from a set of images that have been hand-labelled with 34 2-D vertices demarcating the contours of the lips, jaw and nostrils (illustrated in Figure 1(a)). Generalised procrustes analysis [48] is used to align each set of vertices to the mean jaw shape to remove scale, rotation and translation due to head motion. The 34 pairs of x-y co-ordinates, $\{r(n), r(n+1)\}$, are stacked to create a 68-D vector, $\mathbf{r} = [r(1), r(2), \ldots, r(67), r(68)]$. A set of D_S coefficients, from a principal components analysis (PCA), are then extracted to give parameter vector $\mathbf{s}$ that encodes the shape of the facial pose.

Appearance is modelled with two independent linear models representing the pixel intensities of the inner mouth and jaw areas respectively. The regions are modelled separately since the inner mouth area can change somewhat independently to the rest of the jaw due to the presence and positions of the teeth and tongue. The images are warped to the mean shape and the pixels from each region are extracted. PCA is applied to the stacked pixel intensities to extract appearance vectors, $\mathbf{b}_m$ and $\mathbf{b}_j$, that model the mouth and jaw facial regions with D_O and D_J components, respectively.

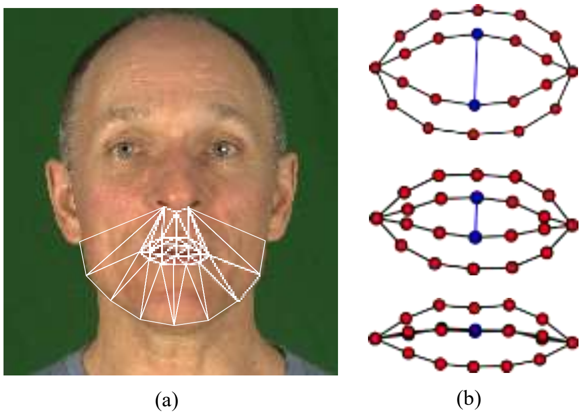

(a) (b)

Fig. 1. *(a) Training image with 34 x-y vertices marked, (b) Example mouth shapes showing the measurement of mouth opening/closure, Δ, which uses the 27th and 32nd co-ordinate pairs.*

Finally, the shape and both appearance features are stacked and another PCA is performed to decorrelate the features. This results in a $D_V = 47$ dimensional vector, $\mathbf{a}_t$, with t the time index of the vector, that combines and compresses the shape and appearance information and retains 98% of the total variation.

Once constructed, the AAM can be fitted to new images by solving for the model parameters [45]. In this way video frames can be tracked and parameterised into feature vectors that encode the position of the jaw and lip area. The videos used in this work (Section VI) have a frame rate of 29.97 fps, so a vector of visual features is created every 33ms.

B. Acoustic features

The acoustic features must encode the characteristics of speech to enable an accurate sequence of visual speech features to be estimated. Given their success in many speech processing applications [49], [50], and in our previous related work [1], the acoustic features are based on mel-frequency cepstral coefficients (MFCCs). Our implementation builds on the ETSI Aurora standard [49] and extracts 20ms frames of speech every 10ms, applies a Hamming window and then computes the power spectrum. A 44-channel mel-spaced filterbank is then applied followed by a log and discrete cosine transform to produce a D_M =44-dimensional MFCC vector, $\mathbf{x}_t$, with no truncation applied. The audio is sampled at 48kHz.

IV. Speaker-independent audio-to-visual mapping

One method to create a sequence of visual features for animation is to first use a speech recogniser to decode an input audio signal into a phoneme sequence. This phoneme sequence can then be mapped to a stream of visual features [6]. The benefit of this approach is that the phoneme sequence is agnostic to speaker identity, so can work for any given input speech, assuming that the speech recogniser is speaker independent. However, this approach requires a large vocabulary continuous speech recogniser which may result in phoneme

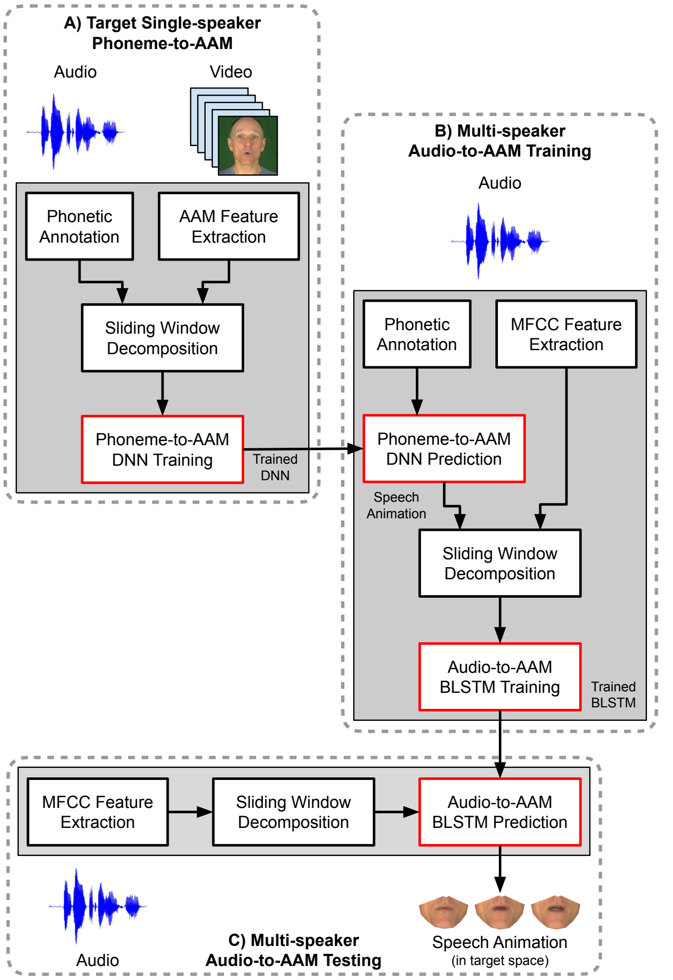


Fig. 2. *Proposed architecture for speaker-independent audio-to-visual mapping.*

errors and will introduce a significant time lag, due to contextual (language) modelling, before producing an output. This makes real-time, online operation impossible. Instead, in this work we propose a speaker-independent architecture to map directly from acoustic features to visual features. An overview of our system is shown in Figure 2.

Training comprises blocks A and B. In the first block (A) we train a speaker-dependent phoneme-to-visual speech model that learns the mapping from phoneme sequences to visual features for a single target speaker. This step requires an audio-visual speech database for the target speaker. In block B, the trained phoneme-to-visual speech model (block A) is used to predict visual features from a phonetically annotated *multi-speaker* database. This step generates synthetic data from which a direct mapping from acoustic features to visual features can be learnt. The result is an audio-to-visual speech model where acoustic features from any speaker can be mapped to AAM parameters representing a single target speaker (block C). Subsequently, the predicted facial motion can be retargeted to other characters if required [6]. The remainder of this section explains the processing blocks in more detail.

A. *Single-speaker phoneme-to-visual speech mapping*

To generate AAM vectors from a phoneme sequence we follow the deep learning approach in [6]. Essentially this begins with an audio-visual speech database taken from a single speaker (target speaker) – we use the KB-2k database described in Section VI. For each training sentence a sequence of visual features, $\mathbf{a}_t$, is extracted using the method described in Section III-A. Accompanying this is a time-aligned phonetic transcription of each sentence which we obtained using an automatic speech recogniser. To improve the transcription, any phoneme or alignment errors were corrected by hand. At each time frame a 41-dimensional binary vector, $\mathbf{p}_t$, is used to indicate which phoneme from the set of 41 Arpabet phonemes is being spoken. To provide context, the sequences of visual and phonetic features are sampled with fixed length overlapping windows, $\mathbf{a}'_t$ and $\mathbf{p}'_t$. The window of visual features, $\mathbf{a}'_t$, comprises K_{VA} AAM vectors while the window of phoneme features, $\mathbf{p}'_t$, provides context over K_P frames to give a $41 \times K_P$ dimensional vector. These windows are both centred at frame t.

Given the training dataset, a feed forward DNN, $h(.)$, is trained to estimate a window of visual features, $\tilde{\mathbf{a}}'_t$, from an input window of phoneme vectors, $\mathbf{p}'_t$, i.e. $\tilde{\mathbf{a}}'_t = h(\mathbf{p}'_t)$. The final sequence of visual features, $\tilde{\mathbf{a}}_t$, is computed by calculating the frame-wise mean of the overlapping predictions. Best performance, in terms of minimising estimation error, was found using three fully connected hidden layers, each with 3000 units and a hyperbolic tangent transfer function. In terms of context window sizes, stacking K_P=11 phonetic features and K_{VA}=5 AAM vectors was found to give best performance. Further details on training can be found in [6].

B. *Multi-speaker audio-to-visual speech mapping*

The system that generates visual features directly from an audio waveform follows the deep learning approach in [23], and maps a sequence of input MFCC vectors to an output sequence of AAM vectors. The approach in [23] was developed for a single speaker and uses an audio-visual speech database (KB-2k) for training that provides both acoustic and visual features. Our approach extends this system to become speaker independent. For this we use acoustic features taken from the multi-speaker TCD-TIMIT database described in Section VI. Obtaining tracked faces, and subsequently visual features, from a large multi-speaker database requires substantial processing and significant human intervention. Instead, we propose synthesising these visual features by phonetically transcribing each training sentence and then predicting the corresponding visual features using the phoneme-to-visual speech mapping method described in Section IV-A. This approach not only removes the need for tracked videos, but also produces visual features that are based on the single speaker (target) AAM used for training the phoneme-to-visual speech model. If independent AAMs were constructed for each speaker in the database, it is unlikely that the set of AAMs would be aligned to each other, making the resulting speaker-specific AAM vectors unsuitable for training. Finally, synthesising visual features in this way allows speech databases to be used that contain only an audio stream, thereby broadening the availability of databases that can be used for training.

For each speaker-independent training sentence the following process is applied:

1) Extract a sequence of MFCC vectors, $\mathbf{x}_t$, using the method described in Section III-B.
2) Transcribe the sentence into a phoneme-level annotation. The TCD-TIMIT database is supplied with these annotations but they could be produced automatically using a speech recogniser.
3) Synthesise a sequence of visual features, $\hat{\mathbf{a}}_t$, from the phoneme transcription using the trained phoneme-to-visual speech model (described in Section IV-A).

As before, to provide context, the sequences of AAM and MFCC features are sampled using fixed length overlapping windows, $\mathbf{a}'_t$ and $\mathbf{x}'_t$. The window of visual features, $\mathbf{a}'_t$, comprises K_{VB} AAM vectors while the window of audio features provides context over K_A frames to give a $D_M \times K_A$ dimensional vector. These are both centred at frame t. From the set of training data, a bi-directional LSTM (BLSTM), $l(.)$, is trained to estimate a window of AAM vectors, $\hat{\mathbf{a}}'_t$, from an input acoustic sequence, $\mathbf{x}'_t$, i.e. $\hat{\mathbf{a}}'_t = l(\mathbf{x}'_t)$. The final sequence of visual features, $\hat{\mathbf{a}}_t$, is computed by calculating the frame-wise mean of the overlapping predictions

Best performance, in terms of minimising mean square error, was found using two pairs of recurrent forward and backward layers. Each contains 256 LSTM cells with peephole connections [51] such that pair $\mathbf{h}^n = [\overrightarrow{\mathbf{h}^n}; \overleftarrow{\mathbf{h}^n}]$. This is followed by a single fully connected layer with 2000 rectified linear units (ReLU) [52] and a linear output layer.

The first LSTM pair, $\mathbf{h}^1$, traverses the full context window of the input acoustic features

$$\overrightarrow{\mathbf{h}^1}, \overleftarrow{\mathbf{h}^1} = [\mathbf{x}_{t-\omega_a}, ..., \mathbf{x}_{t+\omega_a}] \tag{1}$$

whereas the second LSTM pair, $\mathbf{h}^2$, stops traversing after reaching $\mathbf{h}^1_t$ in both forward and backward directions

$$\overrightarrow{\mathbf{h}^2} = [\mathbf{h}^1_{t-\omega_a}, ..., \mathbf{h}^1_t], \quad \overleftarrow{\mathbf{h}^2} = [\mathbf{h}^1_t, ..., \mathbf{h}^1_{t+\omega_a}] \tag{2}$$

where $\omega_a = \lfloor K_A/2 \rfloor$. Only the final output from $\mathbf{h}^2$, representative of the audio-visual alignment at t, is passed through the remaining network. In terms of context window sizes, a sequence of K_A=33 MFCC features (340ms of audio) and K_{VB}=3 visual features (100ms of video) gave best performance. Further details of the audio-to-visual speech model architecture can be found in [23].

C. Speaker-independent audio-to-visual speech mapping

The audio-to-AAM mapping model can now be used to predict AAM features from the audio of any speaker as shown in Block C of Figure 2. From the input audio, MFCC vectors are extracted and sliding window decomposition applied. These are input into the audio-to-AAM model from where a sequence of AAM vectors is output.

V. Perceptual loss function

We generally achieve a low mean square error (MSE) when mapping acoustic features to visual features. However, when visualised, the rendered facial animation is sometimes observed to exhibit poor naturalness. This is most often caused by the mouth not closing or opening in a realistic way and is particularly salient for bilabial closures (/b/, /p/ and /m/) which require the mouth to close fully in order to appear natural. For other speech sounds the lip targets are typically perceptually less important. To address this weakness, we propose a perceptually-based loss function by extending the MSE to focus more on mouth closures and openings.

A. Mean square perceptually-weighted error (MSPE)

The most extreme facial poses occur when AAM coefficients deviate far from their mean values. This is illustrated in Figure 3 which shows AAM coefficients $a(1)$ to $a(10)$ as each is varied from three standard deviations below its mean to three standard deviations above, while setting the remaining coefficients to their mean value. Perceptual errors in animation that are caused by insufficient mouth closure or opening arise from under-prediction of certain AAM coefficients where they are not sufficiently far from their mean value to produce the desired articulation. These extremity values tend to be low probability events as they occur infrequently in training data. It is therefore likely that when extremity values are required, the model will under-predict these (i.e. closer to the mean) rather than attaining the desired value or even over-predicting (i.e. further from the mean).

To enable the model to generate more visually realistic animations for mouth closures and openings, we propose a perceptually motivated loss function that encourages prediction out to extremity values (at both sides of the mean) to make greater mouth closing and opening more likely. This is achieved by introducing a perceptually-based error weighting into the MSE loss function that weights under-prediction and over-prediction of AAM coefficients differently. The conventional MSE loss function, $\mathrm{L}^{\mathrm{MSE}}$, used initially within the audio-to-visual speech mapping in Section IV-B, is defined as,

$$\mathrm{L}^{\mathrm{MSE}} = \frac{1}{T\,D_V} \sum_{t=1}^{T} \sum_{j=1}^{D_V} \big(a_t(j) - \hat{a}_t(j)\big)^2 \tag{3}$$

where $a_t(j)$ and $\hat{a}_t(j)$ represent respectively the jth coefficients of the tth target and estimated visual features, and T and D_V are the number of vectors under test and the number of coefficients in the visual vectors. The proposed mean square perceptually-weighted error loss function, $\mathrm{L}^{\mathrm{MSPE}}$, is defined as,

$$\mathrm{L}^{\mathrm{MSPE}} = \frac{1}{T\,D_V} \sum_{t=1}^{T} \sum_{j=1}^{D_V} \Big[M_t(j)\,\mathrm{ReLU}\big((a_t(j) - \hat{a}_t(j)),\ \alpha\big) + (1 - M_t(j))\,\mathrm{ReLU}\big((\hat{a}_t(j) - a_t(j)),\ \alpha\big)\Big]^2 \tag{4}$$

where the square error calculation within the summation is now divided into two parts according to a mask value. The mask, $M_t(j)$, denotes whether AAM coefficient $a_t(j)$ is above or below its mean value, $\mu_a(j)$, and is defined,

$$M_t(j) = \begin{cases} 1 & \text{for} \quad a_t(j) \geq \mu_a(j) \\ 0 & \text{for} \quad a_t(j) < \mu_a(j) \end{cases} \tag{5}$$

Within the two parts of the squared error calculation a leaky rectified linear unit (ReLU) operator is defined as,

$$\mathrm{ReLU}(x, \alpha) = \begin{cases} \alpha \times x & \text{for} \quad x < 0 \\ x & \text{for} \quad x \geq 0 \end{cases} \quad \text{with } 0 \leq \alpha \leq 1 \tag{6}$$

where α determines the amount of leakage and in this application controls the amount of perceptual weighting within the loss function. For $\alpha < 1$, the contribution made by negative errors (over-prediction) is reduced by a factor α while positive errors (under-prediction) are unchanged. Smaller values of α increase the effect of perceptual weighting while for the case where $\alpha = 1$, no rectification takes place and the loss function is equivalent to MSE. When combined with the mask operator, M, errors are weighted to encourage extremity values.

As an example, when the reference AAM coefficient is greater than its mean value (i.e. $a_t(j) \geq \mu_a(j)$) the loss function encourages over-prediction of AAM coefficients *above* the mean. In this situation the mask value $M_t(j) = 1$ and so the first term in (4) is activated. When the estimated coefficient, $\hat{a}_t(j)$, is over-predicted the resulting error $(a_t(j) - \hat{a}_t(j))$ is negative and is therefore reduced by the ReLU operator which in turn encourages over-prediction within the loss function. When the coefficient is under-predicted, the error is positive and unchanged by the ReLU operator. Conversely, when the reference AAM coefficient is less than its mean value (i.e. $a_t(j) < \mu_a(j)$) the loss function encourages over-prediction *below* the mean by activating the second term in (4). When the estimated coefficient, $\hat{a}_t(j)$, is over-predicted (i.e. lower than the reference) the error $(\hat{a}_t(j) - a_t(j))$ is negative and is therefore reduced by the ReLU operator which encourages over-prediction below the mean value. For under-predicted estimates, the error is positive and remains unchanged.

B. Improving mouth closure and opening

The perceptually-weighted loss function, $\mathrm{L}^{\mathrm{MSPE}}$, in (4) can be evaluated over all visual features during model training. However, the $\mathrm{L}^{\mathrm{MSPE}}$ loss function will increase MSE across all AAM coefficients and may introduce artefacts that were previously not present. To reduce this unwanted effect, the $\mathrm{L}^{\mathrm{MSPE}}$ loss function is adapted to affect just mouth closures and openings (the most perceptually salient errors), L^{CO}, or just mouth closures, L^{C}, by targeting only those AAM coefficients that explicitly represent mouth closure and opening. The conventional MSE calculation is applied to the remaining coefficients. These two perceptual loss functions, L^{CO} and L^{C}, are defined,

$$\begin{aligned} \mathrm{L}^{\mathrm{CO}} = \frac{1}{T\,D_V} \sum_{t=1}^{T} \sum_{j=1}^{D_V} \Big[& I(j)\, M_t(j)\, \mathrm{ReLU}\big((a_t(j) - \hat{a}_t(j)),\, \alpha\big) \\ & + I(j)\, (1 - M_t(j))\, \mathrm{ReLU}\big((\hat{a}_t(j) - a_t(j)),\, \alpha\big) \\ & + (1 - I(j))\, (a_t(j) - \hat{a}_t(j))\Big]^2 \end{aligned} \tag{7}$$

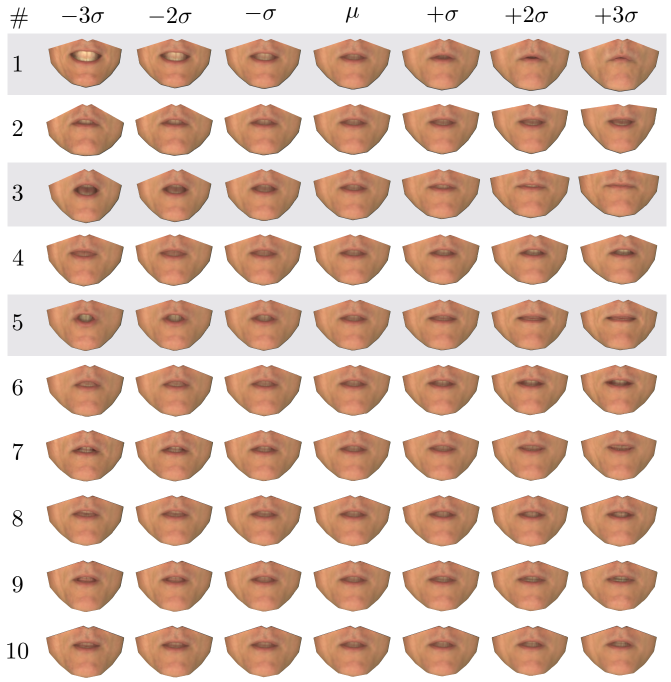


Fig. 3. *The first ten AAM coefficients $a(1)$ to $a(10)$ as each is varied by three standard deviations below and above its mean value.*

$$\begin{aligned} \mathrm{L}^{\mathrm{C}} = \frac{1}{T\,D_V} \sum_{t=1}^{T} \sum_{j=1}^{D_V} \Big[& I(j)\, M_t(j)\, \mathrm{ReLU}\big((a_t(j) - \hat{a}_t(j)),\, \alpha\big) \\ & + I(j)\, (1 - M_t(j))\, (a_t(j) - \hat{a}_t(j)) \\ & + (1 - I(j))\, (a_t(j) - \hat{a}_t(j))\Big]^2 \end{aligned} \tag{8}$$

A second mask, $I(j)$, indicates whether AAM coefficient $a(j)$ is within the subset of coefficients, A, that represent mouth closure or opening, defined as,

$$I(j) = \begin{cases} 1 & \text{for} \quad a(j) \in A \\ 0 & \text{for} \quad a(j) \notin A \end{cases} \tag{9}$$

Considering loss function, L^{CO}, this addresses both mouth closures and mouth openings. The first two terms in the summation are the perceptually-weighted error which is applied to the subset of AAM coefficients, A, that represent mouth closures and openings. The final term computes an unweighted error for the remaining coefficients that are not in A. The second loss function, L^{C}, is designed to weight only mouth closures, as under-articulated closures are perceptually worse than under-articulated openings, particularly for bilabial phonemes. This is achieved by reducing the second term in the summation, which corresponds to mouth openings, to an unweighted error.

Figure 3 shows that coefficients $a(1)$, $a(3)$ and $a(5)$ contribute most to mouth closing and opening. We therefore experiment with generating animations using each of these modes individually as $A_1 = \{a(1)\}$, $A_3 = \{a(3)\}$ and $A_5 = \{a(5)\}$ and then combining all three to give $A_{135} = \{a(1),\ a(3),\ a(5)\}$. The effectiveness of the two perceptually weighted loss functions is evaluated in Section VI, where models trained with either L^{CO} or L^{C} are compared against the unweighted, $\mathrm{L}^{\mathrm{MSE}}$, loss function.

VI. Results

We evaluate the proposed speaker-independent animation system and perceptual loss functions using both objective and subjective tests. We first investigate the efficacy of the proposed architecture and loss functions on a single-speaker dataset for which we have ground truth visual features. This allows us to perform an initial refinement of the system. Next, we evaluate on a multi-speaker dataset. Finally, we report results on subjective testing.

The single-speaker setting is evaluated using the KB-2k audio-visual speech dataset which comprises an American male uttering 2,084 phonetically balanced sentences in a neutral style [12]. The audio sampling frequency is 48kHz while the video is captured at 29.97fps. This dataset has been phonetically annotated using a speech recogniser, and then manually hand-corrected to ensure correctness. Sentences are split randomly into training, validation and test sets containing 1884, 100 and 100 sentences each. The multi-speaker system is evaluated using the TCD-TIMIT dataset which comprises 55 speakers each uttering approximately 98 sentences [53]. This is partitioned into 39 speakers for training (20 male and 19 female), 6 for validation (3 male and 3 female) and 10 for testing (5 male and 5 female). The audio and video were sampled at the same rates as for the KB-2k dataset. Phonetic annotations are supplied with this dataset and were created using forced alignment, based on the reference phoneme sequence.

A. Single-speaker analysis

This section presents experimental results using the single speaker from the KB-2k dataset for testing which is accompanied by ground truth visual features extracted from the original video. We first measure the baseline performance using the conventional MSE loss function before examining the effect of the perceptually-weighted loss functions applied to various combinations of AAM coefficients.

The objective evaluations use the MSE from (3) and global variance (GV), which is calculated as

$$\mathrm{GV} = \frac{1}{T\,D_V}\sum_{t=1}^{T}\sum_{j=1}^{D_V}\left(\hat{a}_t(j) - \mu_{\hat{a}}(j)\right)^2 \tag{10}$$

where $\hat{a}_t(j)$ represents the jth coefficient of the tth estimated visual feature, and $\mu_{\hat{a}(j)}$ is the mean of the estimate of the jth visual feature. T and D_V are, respectively, the number of vectors under test and the number of coefficients in the visual features. The MSE is a commonly used metric but we also include global variance as we find this to correlate better with the perceptual naturalness of the animation, with higher GV generally showing higher levels of articulation [54].

We also measure the effect of using perceptually-weighted loss functions on the predicted animation. Whilst MSE and GV indicate the general accuracy of visual features against ground truth features, they do not explicitly measure the extent of mouth closure. To evaluate mouth closure explicitly we calculate the distance, Δ, between x-y co-ordinates taken from the pair of 2-D vertices that correspond to the middle of the top inner lip and middle of the bottom inner lip, shown in Figure 1(b). These vertices correspond to elements $\{r_t(63), r_t(64)\}$ and $\{r_t(53), r_t(54)\}$. The mouth opening for the reference (target) and estimated mouth animations, Δ_t^{REF} and Δ_t^{EST}, are then calculated as

$$\Delta_t^{REF} = \sqrt{(r_t(63) - r_t(53))^2 + (r_t(64) - r_t(54))^2} \tag{11}$$

$$\Delta_t^{EST} = \sqrt{(\hat{r}_t(63) - \hat{r}_t(53))^2 + (\hat{r}_t(64) - \hat{r}_t(54))^2} \tag{12}$$

Using these reference and estimated mouth openings, the mouth error for each animated frame is calculated as a simple difference between the two, with the sign of the error retained. The mean mouth error, ε, is then computed as the average error across all T frames in the test data as

$$\varepsilon = \frac{1}{T}\sum_{t=1}^{T}(\Delta_t^{REF} - \Delta_t^{EST}) \tag{13}$$

Values of ε close to zero show good estimation of the amount of mouth opening. Negative values of ε indicate that the estimated mouth opening is larger than the reference mouth and so not closed enough. Positive values indicate that the estimated mouth opening is smaller than the reference and not open enough.

1) Baseline evaluation: We begin by evaluating the performance of three baseline methods that use the conventional MSE loss function. This is shown in Table I and considers the MSE, GV and mouth error metrics, split into bilabial closure phonemes (/b/, /p/ and /m/) and non-bilabial closure phonemes (i.e. the set of phonemes excluding /b/, /p/ and /m/). First, we include the baseline phoneme-to-visual speech mapping method, based on [6] and described in Section IV-A (Phoneme-to-AAM). This uses hand corrected phoneme annotations and so represents an ideal system. Second is a speaker-dependent system based on [23] that maps directly from acoustic features to AAM features using the ground-truth data (Audio-to-AAM). This is trained on the single speaker dataset and, although not appropriate for later speaker-independent operation, it serves as a useful baseline to evaluate the error introduced by mapping directly from audio compared with from phonemes. The third system (Audio-to-AAM (Synth)) also maps from audio to AAM features but is now trained on visual features that have been synthesised by the Phoneme-to-AAM system and follows the three-stage framework proposed in Section IV-B. These three systems are evaluated against ground truth visual features extracted from the single speaker in the KB-2k dataset. The fourth entry in Table I shows results from the proposed synthetically trained model (Audio-to-AAM (Synth)) but with its outputs evaluated against synthesised test data visual features produced by the Phoneme-to-AAM system, as these are generated in the same way as the targets used during its training.

Table I shows that the Phoneme-to-AAM mapping architecture has larger MSE than the Audio-to-AAM mapping, but produces more accurate mouth closures. The AAM vectors produced from this system are subsequently used to train the Audio-to-AAM (Synth) system (Section IV-B), so observing good accuracy here is important. In practice, errors for this method would be larger as the hand corrected phoneme annotations would be replaced by those generated from an

TABLE I

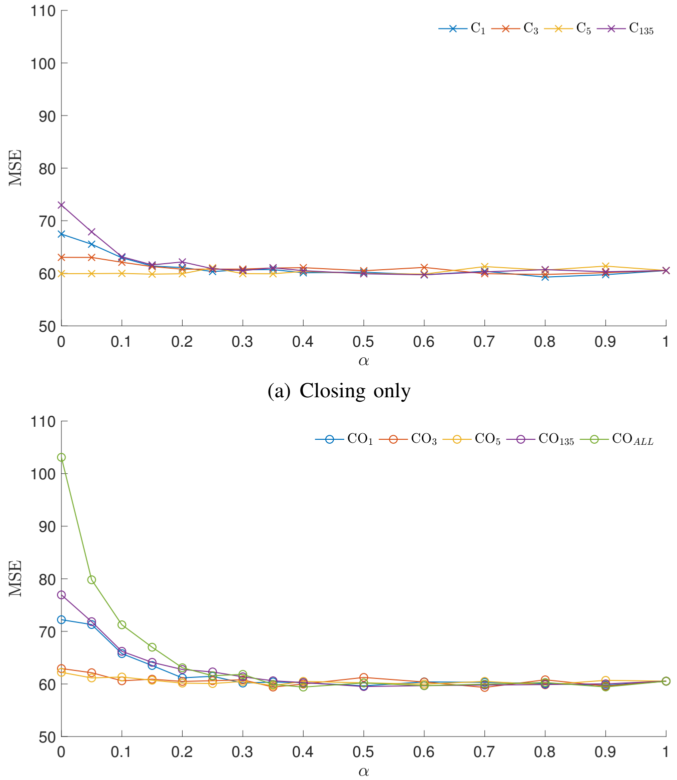
BASELINE PERFORMANCE. MEAN SQUARE ERROR, GLOBAL VARIANCE AND MOUTH ERROR FOR BILABIAL CLOSURES AND NON-BILABIAL CLOSURES FOR THE PHONEME-TO-AAM, AUDIO-TO-AAM TRAINED ON GROUND TRUTH DATA AND AUDIO-TO-AAM TRAINED ON SYNTHESISED DATA SYSTEMS. THE FINAL ROW IS EVALUATED AGAINST SYNTHESISED TEST DATA.

System	MSE	GV	ε (bilab)	ε (non-bilab)
Phoneme-to-AAM [6]	55.49	93.29	-0.93	0.11
Audio-to-AAM [23]	46.39	103.18	-1.04	0.84
Audio-to-AAM (Synth)	60.53	81.77	-2.77	0.17
Audio-to-AAM (Synth) against Phoneme-to-AAM	*13.48*	*81.77*	*-1.84*	*0.06*

automatic speech recogniser and likely contain errors in the phoneme labels and their start and end times. In terms of mouth error, ε, all methods show insufficient mouth closure ($\varepsilon < 0$) for the bilabial closure phonemes (/b/, /p/ and /m/) with the Audio-to-AAM (Synth) method performing worst. For non-bilabial closures, all methods predict slightly too much mouth closure ($\varepsilon > 0$) although the visual consequence is likely to be perceptually less serious compared to animating bilabial closures. When evaluating the Audio-to-AAM (Synth) system against visual features produced from the Phoneme-to-AAM system (row 4), a low error is achieved which shows the mapping to be effective.

2) Evaluation of perceptual loss functions: We evaluate the effect of using the perceptually-weighted loss to train the Audio-to-AAM mapping using the single speaker dataset. Figures 4(a) and 5(a) show MSE and GV for the closing-only loss function, L^{C}, applied to different sets of AAM coefficients as α is varied from 0 to 1 to change the perceptual weighting. Specifically, from the observations in Section V-B, the sets of AAM coefficients investigated are $a(1)$, $a(3)$, $a(5)$ and $a(1,3,5)$, which are shown as lines C_1, C_3, C_5 and C_{135}. Similarly, figures 4(b) and 5(b) show MSE and GV for the closing and opening loss function, L^{CO} applied to the same sets of AAM coefficients plus an additional set that includes all coefficients. These are shown as lines CO_1, CO_3, CO_5, CO_{135} and CO_{ALL}. At α=1 the perceptual terms in the loss function are inactive, and the loss is reduced to the standard MSE loss function as shown in row 3 of Table I. Reducing α increases the perceptual contribution within the loss functions which increases the MSE and global variance. This rise is expected as the perceptual parts of the loss function are not designed to minimise the MSE of the estimated visual features.

Figure 6 shows mouth error, ε, computed over just the set of bilabial closure phonemes (/b/, /p/ and /m/) for the closing, and closing and opening loss functions, as α is varied from 0 to 1. Using the MSE-only loss function (i.e. α=1), the mouth error, ε, is -2.77 which corresponds to row 3 of Table I. This indicates that the mouth is generally not closed enough for the bilabial closures and when animated does reveal poor realism in many cases. As α is reduced, the increasing contribution of the perceptual component of the loss functions improves mouth closure as evidenced by the increasing value of ε from -2.77. Considering the effect of individual AAM coefficients, a small increase in mouth closure is observed for $a(3)$ and $a(5)$, while $a(1)$ introduces much more mouth closure. Combining multiple AAM coefficients also has large effect as in C_{135}, CO_{135} and CO_{ALL}. Informal observation of the resulting animations of predicted bilabial closure phonemes confirms that mouth closure is now achieved more often and realism has been improved.



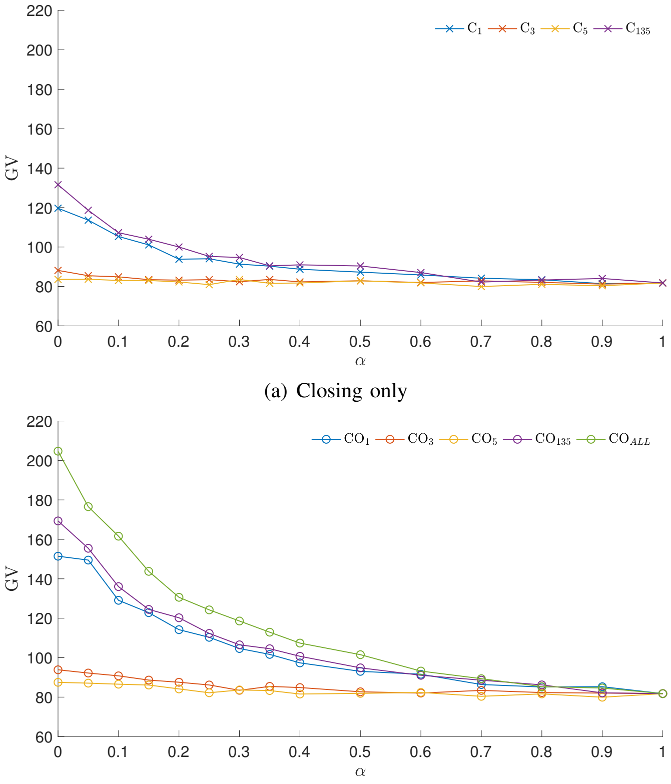
(a) Closing only

(b) Closing and Opening

Fig. 4. *Mean square error of audio-to-visual speech estimation as the perceptual weighting, α, is varied from 0 to 1 for closing only and closing and opening loss functions.*

The perceptual loss functions make no distinction as to whether the input speech is a bilabial closure or not, and consequently mouth closure is also affected for non-bilabial closure phonemes. Figure 7 shows mouth error, ε, for the same set of loss functions and values of α, but now calculated over the set of non-bilabial phonemes (i.e. excluding /b/, /p/ and /m/). Using the MSE loss function (α=1), the mouth error, ε, is 0.17, which is much closer to zero error than for bilabial closures and indicates a mouth opening that is slightly too small. As α is reduced, which increases the perceptual part of the loss functions, the closing-only loss functions introduce more mouth closure, while the opening/closing loss functions have less effect on mouth error.

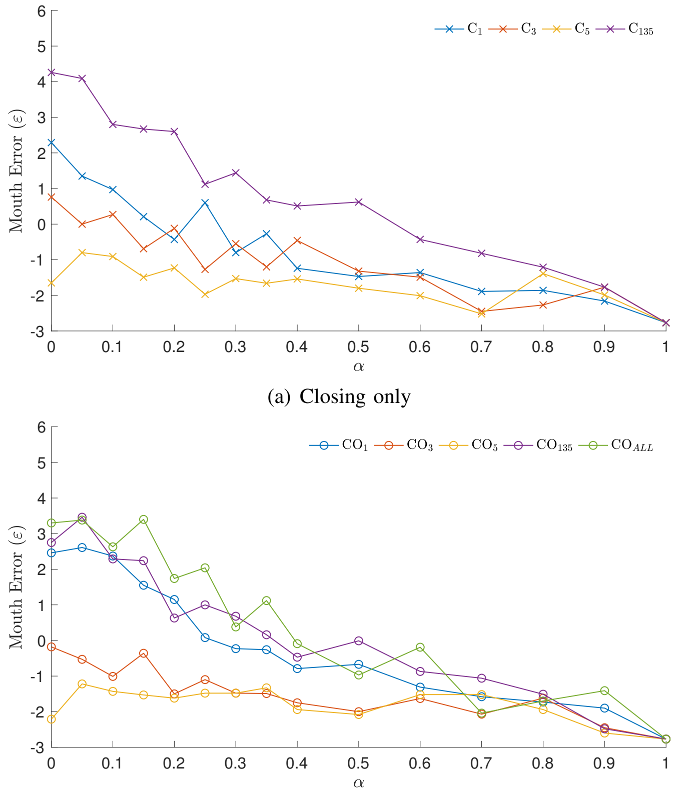
Ideally, mouth closures should increase for bilabial closure phonemes and not change for non-bilabial closures. Practically, however, a trade-off must be made as well as consideration of the MSE and GV metrics. Consequently, for further multi-speaker testing we select perceptual loss functions C_1, CO_1, C_{135}, CO_{135} and CO_{ALL} as these give the best balance across both objective scores and from our informal observations of the animation.



(a) Closing only

(b) Closing and Opening

Fig. 5. *Global variance of audio-to-visual speech estimation as the perceptual weighting, α, is varied from 0 to 1 for closing only and closing and opening loss functions.*



(a) Closing only

(b) Closing and Opening

Fig. 6. *Mouth error of audio-to-visual speech estimation for bilabial closure phonemes (/b/, /p/, /m/) for different perceptual weightings. Negative error corresponds to mouth opening too large, positive error corresponds to mouth opening too small.*

B. Multi-speaker analysis

We evaluate the speaker-independent audio-to-visual speech mapping using the TCD-TIMIT multi-speaker database. Ten speakers are used for testing and each contributes 98 sentences. With this speaker-independent dataset there are no ground truth visual features that can be used for evaluation, so instead AAM vectors produced using the phoneme-to-visual speech system in Section IV-A are used as the reference data for analysis. Following objective tests, a set of subjective tests are then performed to compare the naturalness of animations for different perceptual loss functions.

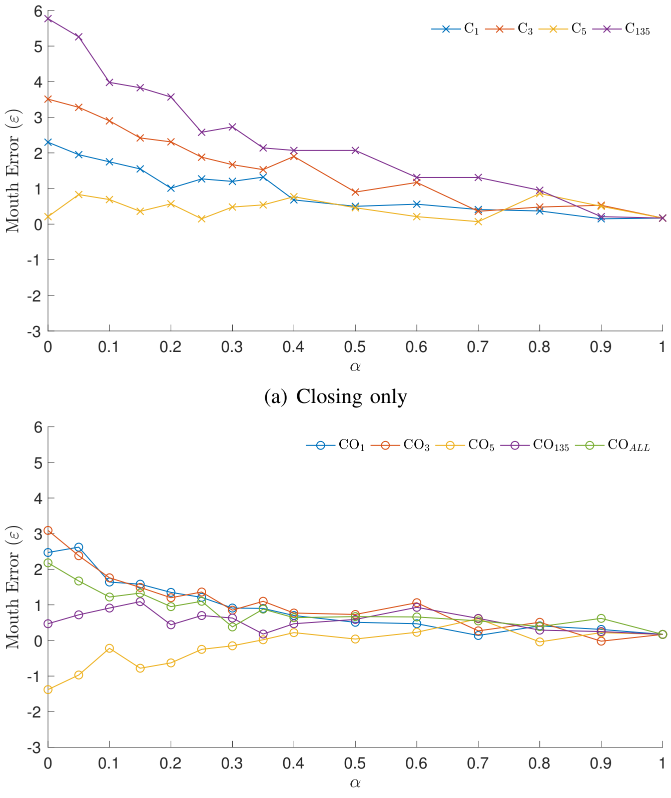
1) Objective evaluation: Figure 8 shows MSE, global variance and mouth error for bilabial closures and non-bilabial closures for the five perceptual loss functions that were found to perform best in Section VI-A2 (C_1, CO_1, C_{135}, CO_{135} and CO_{ALL}) and are evaluated across α from 0 to 1.

Considering mouth error first, for bilabial closures insufficient mouth opening is observed with the standard MSE loss function (i.e. α=1 where ε=-1.35). For non-bilabial closures, the mouth error is positive indicating too much closure with ε=1.28. These results are similar to those observed with the single speaker analysis in Section VI-A. When the perceptual loss function is applied to all coefficients for mouth openings and closings, CO_{ALL}, we observe greater mouth closure across both bilabial closures and non-bilabial closures as α is reduced from 1 to 0. This loss function also gives the largest increases in MSE and GV across all combinations of loss functions and AAM coefficients.

When the parameters influenced by the perceptual loss function are reduced to CO_{135}, only a small increase in mouth closure is introduced as α reduces from 1.0 to around 0.4. Below this the mouth closure for bilabial closures increases substantially while for non-bilabial closures the change is less, which is desirable. Removing the opening component of the loss function to give C_{135} causes too much mouth closure, particularly for non-bilabial closures. Restricting the loss function to openings and closures applied to the first AAM coefficient only, CO_1, has much less effect on mouth closure and makes only a small gain on bilabial closures. Removing the opening part of the loss function to give C_1, follows a similar trend and both are highly variable for small changes in α.

In terms of MSE, this increases as α is reduced for all loss functions, as was observed for single speaker testing, with CO_{ALL} being affected most which is due to the perceptual loss function being applied across all visual features. Global variance exhibits a similar trend, with larger increases found with CO_{ALL} and CO_{135}.

2) Subjective analysis: Finally, we measure the naturalness of our proposed system using subjective tests. Specifically, we consider the four best system configurations determined by the objective tests and from informal observation of animations. These are systems CO_{ALL} and CO_{135} which we investigate



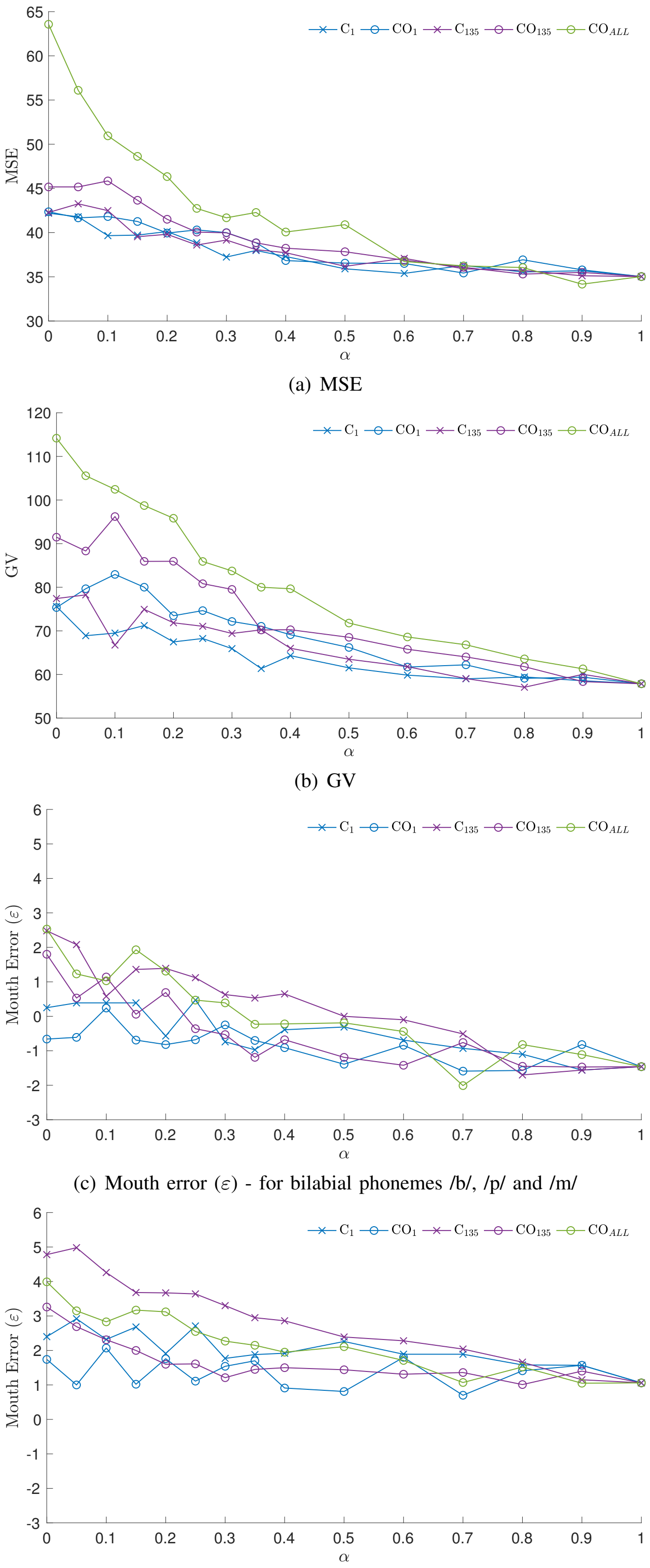
(a) Closing only

(b) Closing and Opening

Fig. 7. *Mouth error of audio-to-visual speech estimation for non-bilabial closure phonemes (excluding /b/, /p/, /m/) for different perceptual weightings. Negative error corresponds to mouth opening too large, positive error corresponds to mouth opening too small.*

each with perceptual weight values of α=0.4 and α=0.1. For comparison, the system using the standard MSE loss function is included (i.e. when α=1.0) and also the Phoneme-to-visual speech system described in Section IV-A that is used to synthesise target visual features for training. These six systems are evaluated using a complete set of pairwise preference tests, yielding 15 combinations in total. In each session, subjects are played three examples from each pair of configurations, all in a random order, giving a total of 45 test comparisons. For each comparison, a video is played that shows the pair of animations side-by-side and subjects are asked to select which one they think is more natural. The original audio accompanies the video. Participants are able to view the videos as many times as they wish. Each of the 45 comparisons takes its sentence at random from one of the 10 different test speakers from the TCD-TIMIT dataset. Before the test, subjects are played three videos which together show all six systems, taken at random. This forms an initialisation phase that subjects are not told about, and the results from these are discarded. The tests are performed using a web interface with 30 subjects participating.

Table II shows the results of the preference tests where each entry shows the percentage preference of System A over System B. A Chi-squared test was also performed and where the result is statistically significant (at 95%) an asterisk is used to indicate this. The Phoneme-to-visual speech system is most preferred by subjects, although this is a somewhat



(a) MSE

(b) GV

(c) Mouth error (ε) - for bilabial phonemes /b/, /p/ and /m/

(d) Mouth error (ε) - for non-bilabial phonemes

Fig. 8. *Speaker-independent MSE, GV and mouth error for varying loss functions evaluated against the phoneme-to-visual speech predictions as perceptual weighting, α, is varied.*

TABLE II

Subjective preference scores comparing animations produced from six different methods of estimating visual features. An asterisk indicates the result is statistically significant at 95% confidence.

System B System A	P-to-AAM	MSE(1.0)	$\mathrm{CO}_{ALL}(0.4)$	$\mathrm{CO}_{135}(0.4)$	$\mathrm{CO}_{ALL}(0.1)$	$\mathrm{CO}_{135}(0.1)$
Phoneme-to-AAM	–	75.79*	64.04*	56.52	53.93	47.67
$\mathrm{MSE}(\alpha = 1.0)$	24.21*	–	35.96*	36.67*	22.34*	31.52*
$\mathrm{CO}_{ALL}(\alpha = 0.4)$	35.96*	64.04*	–	53.93	39.56*	32.58*
$\mathrm{CO}_{135}(\alpha = 0.4)$	43.48	63.33*	46.07	–	47.78	35.56*
$\mathrm{CO}_{ALL}(\alpha = 0.1)$	46.07	77.66*	60.44*	52.22	–	54.02
$\mathrm{CO}_{135}(\alpha = 0.1)$	52.33	68.48*	67.42*	64.44*	45.98	–

artificial result as the phoneme annotations and timings are generated using forced alignment which would not be available in practice. However, this is an important result as it is this system that is used to synthesise the training data for the subsequent systems developed in Section IV. The least preferred system uses the standard MSE loss function which has no perceptual weighting (i.e. α=1.0). Applying the perceptual loss function improves naturalness in all cases. Considering the effect of the perceptual weighting, the two systems with higher weighting, CO_{135}(α=0.1) and CO_{ALL}(α=0.1), are both preferred over the two with lower weighting, CO_{135}(α=0.4) and CO_{ALL}(α=0.4). Comparing the effect of applying the loss function to all AAM coefficients, CO_{ALL}, to just those related to mouth closing and opening, CO_{135}, reveals little difference in preference. This indicates that whichever loss function is used, it is important to ensure that AAM coefficients $a(1)$, $a(3)$ and $a(5)$ are perceptually weighted.

To examine further the realism of the animated mouth, the upper plot of Figure 9 shows the mouth openness, Δ^{EST}, for the sentence 'the paperboy bought two apples and three ices' from the Phoneme-to-AAM system and the Audio-to-AAM system using the MSE loss function (i.e. α=1.0) and the CO_{135} and CO_{ALL} loss functions (with α=0.1). Considering instances of bilabial closures, for example the instances of phoneme /p/ at frames 3, 10 and 40, both CO_{135} and CO_{ALL} achieve tighter closures compared with the Phoneme-to-AAM and MSE systems. The result of this can be seen in the lower plot of Figure 9 which shows the predicted animations and lip shapes for the frames corresponding to the word 'paperboy' (frames 3 to 20) for each of the four conditions. Animations generated using perceptual loss functions (CO_{ALL} and CO_{135}) can both be seen to give tighter closures for the /p/ and /b/ phonemes in the word.

VII. Discussion

This work has shown that our proposed three-stage framework is able to generate realistic animations from a range of speakers. However, an aim of this work is not just to demonstrate an effective end-to-end system, but also to provide a framework that is modular by design. To validate this outcome, we explore replacing the Audio-to-AAM module in the third stage of our framework with a different architecture. Specifically, we substitute our bi-directional LSTM (BLSTM) with an architecture based on [40] that takes in raw audio, extracts spectrograms, and applies these first to a series of CNN layers to extract features before applying them to an LSTM and dense layers to predict AAM parameters.

TABLE III

Evaluation of the proposed three-stage framework using three variants of the audio-to-AAM module: i) proposed BLSTM with perceptual loss function (CO_{135}(α=0.1)), ii) baseline BLSTM method with MSE loss function [23] and iii) CNN-LSTM method [40]. Evaluation metrics are mean square error, global variance, mouth error for bilabial closures and non-bilabial closures and the number of model parameters.

			ε	ε	
System	MSE	GV	(bilab)	(non-bilab)	#Params
BLSTM (CO_{135})	45.85	96.21	1.14	2.31	3.5M
BLSTM (MSE) [23]	35.03	57.87	-1.46	1.06	3.5M
CNN-LSTM [40]	53.74	33.41	-1.16	3.54	8.4M

This allows us to examine the effectiveness of our framework by comparing three different modules for the final audio-to-AAM stage. First is our proposed method that uses the perceptual loss function and BLSTM architecture, specifically configuration CO_{135}(α=0.1). Second is the baseline MSE loss function and BLSTM architecture [23]. Third is the CNN-LSTM architecture [40]. These are evaluated using the same training and testing configurations as in Section VI-B. Table III shows the MSE, GV and mouth error, ε, split into bilabial and non-bilabial phonemes, for each of the methods with the three-stage framework.

These results confirm that our three-stage framework remains effective when a method in one module is changed for another. The BLSTM with the conventional MSE loss function is essentially loss function CO_{ALL}(α=1.0), and this has already been shown to be inferior to the perceptually weighted loss function, both objectively in Section VI-B1 and subjectively in Section VI-B2. Comparing against the CNN-LSTM method shows this to achieve lower performance. This we attribute to its use of an LSTM, whereas our approach uses a bi-directional LSTM that allows temporal structure to be modelled more effectively which is important when generating animations parameters. As a further comparison, we also show in the final column of Table III the total number of parameters in each model. The CNN-LSTM method has over twice the number of parameters as the BLSTM methods, with the increase arising from the CNN stages.

Other deep learning model architectures could equally be substituted into our proposed framework at either the phoneme-to-AAM stage or the audio-to-AAM stage. For

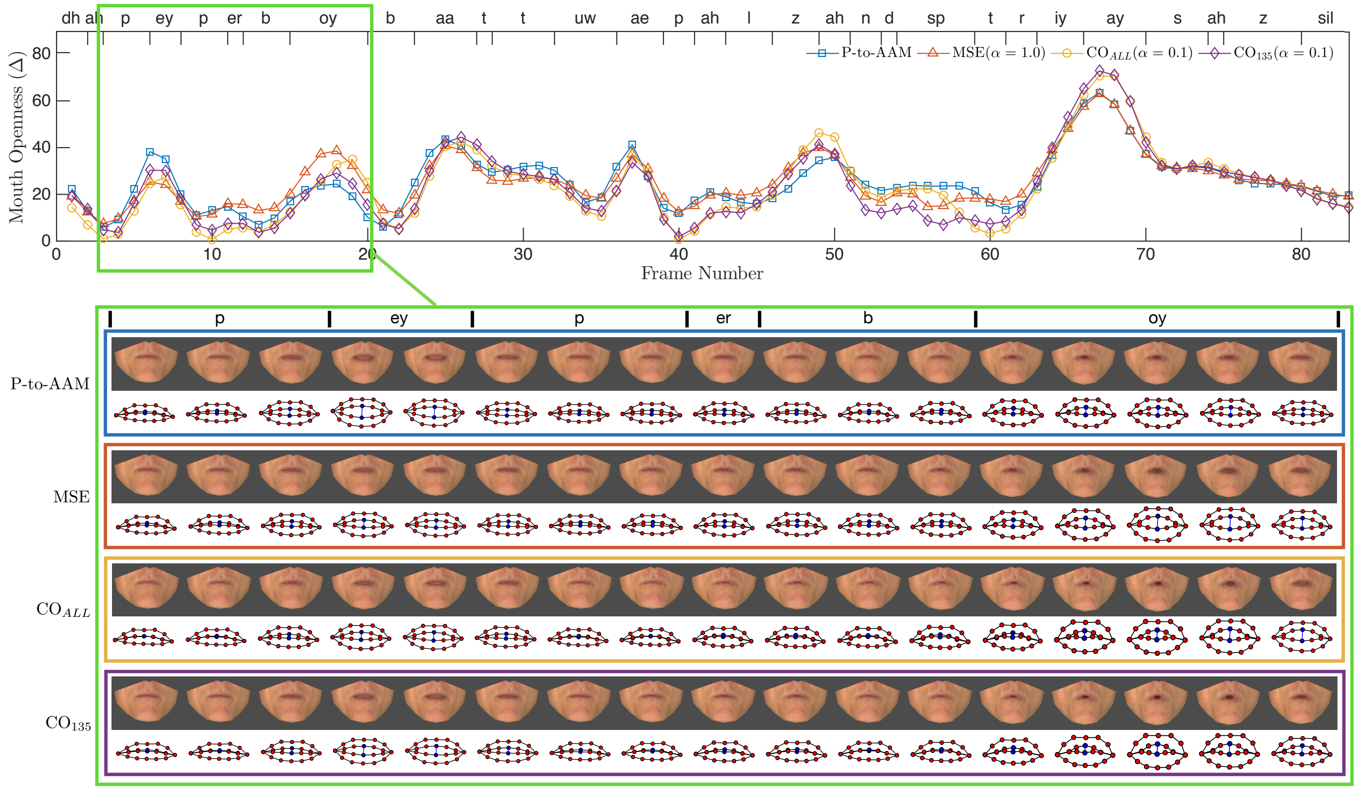


Fig. 9. *Mouth openness (top) and predicted animation (bottom) corresponding to the test sentence 'the* ***paperboy*** *bought two apples and three ices' from the Phoneme-to-AAM system and the Audio-to-AAM system using the MSE loss function (i.e. α=1.0) and the* CO$_{135}$ *and* CO$_{ALL}$ *loss functions (with α=0.1). Aligned phoneme labels are given above each plot.*

example, GANs, VAEs and the other approaches discussed in Section II could all be configured to estimate the synthetic visual targets from the speaker-independent acoustic information. Additionally, in many cases, our proposed perceptually-weighted loss function could also replace directly the existing objective functions in supervised learning techniques.

VIII. Conclusion

In this work we have introduced a three-stage speaker-independent speech animation framework that can be trained using a single-speaker audio-visual speech dataset and a multi-speaker audio-only dataset and is capable of operating in real-time. Specifically, we have proposed a speech animation pipeline that uses synthetic visual parameters, estimated from a speaker-independent phoneme-to-visual speech model, to train a multi-speaker audio-to-visual speech model. Furthermore, we describe a novel perceptually-weighted loss function that improves the perceived naturalness particularly during the production of bilabial closures. The delay introduced through the look-ahead needed for windowing input acoustic features is 170ms which is below the 200ms ITU recommendation for real-time services [22]. Furthermore, recent work has shown that asymmetric windows can further reduce look-ahead lags [23].

The proposed system was evaluated first within a single speaker setting and then within a multi-speaker setting. Without a perceptually-weighted loss, the facial animation sometimes exhibited poor realism even when attaining generally low MSE. We observed that this effect was more perceptually significant when mouth closures were under-articulated, particularly when animating the bilabial closure sounds of /b/, /p/ and /m/. To address this weakness, we proposed a perceptually-based loss function that mitigates under-prediction errors and subsequently improves the animation of bilabial closures. Objective measures confirmed that the new perceptually-weighted loss function was able to improve the accuracy of bilabial closures. Although this came at a cost to the overall MSE, subjective evaluations confirmed that the animations generated were significantly preferred over those using the standard MSE loss. The best performing audio-to-visual speech models (CO$_{ALL}$ and CO$_{135}$ with α=0.1) were trained using the perceptually weighted loss function applied to both mouth openings and closures, L^{CO}, and showed results comparable to the phoneme-to-visual speech system. As a final evaluation, we showed that our three-stage framework is modular, by replacing our BLSTM audio-to-AAM model with a CNN-LSTM taken from [40]. This worked within our framework, although objective evaluation showed our proposed perceptually-weighted BLSTM to perform better.

Acknowledgements

This work was supported by the Engineering and Physical Research Council (Grant numbers EP/M014053/1 and EP/S001816/1). The research presented in this paper was carried out on the High Performance Computing Cluster supported by the Research and Specialist Computing Support service at the University of East Anglia.

References

[1] S. Taylor, A. Kato, I. Matthews, and B. Milner, "Audio-to-visual speech conversion using deep neural networks," in *Interspeech*, 2016, pp. 1482–1486.

[2] T. Karras, T. Aila, S. Laine, A. Herva, and J. Lehtinen, "Audio-driven facial animation by joint end-to-end learning of pose and emotion," *ACM Trans. on Graphics*, vol. 36, no. 4, pp. 94:1–94:12, 2017.

[3] S. Deena, S. Hou, and A. Galata, "Visual speech synthesis using a variable-order switching shared Gaussian process dynamical model," *IEEE Transactions on Multimedia*, vol. 15, no. 8, pp. 1755–1768, 2013.

[4] G. Tian, Y. Yuan, and Y. 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)*, 2019, pp. 366–371.

[5] A. Thangthai, B. Milner, and S. Taylor, "Visual speech synthesis using dynamic visemes, contextual features and DNNs," in *Interspeech*, 2016, pp. 2458–2462.

[6] S. Taylor, T. Kim, Y. Yue, M. Mahler, J. Krahe, A. Rodriguez, J. Hodgins, and I. Matthews, "A deep learning approach for generalized speech animation," *ACM Trans. on Graphics*, vol. 36, no. 4, pp. 270–287, Jul. 2017.

[7] O. Schreer, R. Englert, P. Eisert, and R. Tanger, "Real-time vision and speech driven avatars for multimedia applications," *IEEE Transactions on Multimedia*, vol. 10, no. 3, pp. 352–360, 2008.

[8] M. Cohen and D. Massaro, "Modeling coarticualtion in synthetic visual speech," in *Models and Techniques in Computer Animation*, N. Thalmann and T. D, Eds. Springer-Verlag, 1994, pp. 141–155.

[9] T. Ezzat, G. Geiger, and T. Poggio, "Trainable videorealistic speech animation," in *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '02, 2002, pp. 388–398.

[10] C. Charalambous, Z. Yumak, and A. F. van der Stappen, "Audio-driven emotional speech animation for interactive virtual characters," *Computer Animation and Virtual Worlds*, vol. 30, pp. 1–11, 2019.

[11] C. Bregler, M. Covell, and M. Slaney, "Video rewrite: Driving visual speech with audio," in *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '97, 1997, pp. 353–360.

[12] S. L. Taylor, M. Mahler, B.-J. Theobald, and I. Matthews, "Dynamic units of visual speech," in *Eurographics/ACM SIGGRAPH Symposium on Computer Animation*, 2012, pp. 275–284.

[13] W. Mattheyses, L. Latacz, and W. Verhelst, "Automatic viseme clustering for audiovisual speech synthesis," in *Proceedings of Interspeech*, 2011, pp. 2173–2176.

[14] R. Anderson, B. Stenger, V. Wan, and R. Cipolla, "Expressive visual text-to-speech using active appearance models," in *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 3382–3389.

[15] B. Fan, L. Wang, F. K. Soong, and L. Xie, "Photo-real talking head with deep bidirectional LSTM," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4884–4888.

[16] T. Kim, Y. Yue, S. Taylor, and I. Matthews, "A decision tree framework for spatiotemporal sequence prediction," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015, pp. 577–586.

[17] K. Vougioukas, S. A. Center, S. Petridis, and M. Pantic, "End-to-end speech-driven realistic facial animation with temporal gans," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 37–40.

[18] H. X. Pham, Y. Wang, and V. Pavlovic, "End-to-end learning for 3d facial animation from speech," in *Proceedings of the 2018 on International Conference on Multimodal Interaction*. ACM, 2018, pp. 361–365.

[19] C. Ding, L. Xie, and P. Zhu, "Head motion synthesis from speech using deep neural networks," *Multimedia Tools and Applications*, vol. 74, no. 22, pp. 9871–9888, 2014.

[20] S. Taylor, B. Milner, and A. Kato, "Audio-to-visual speech conversion using deep neural networks," in *Proceedings of Interspeech*, 2016, pp. 1482–1486.

[21] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina *et al.*, "State-of-the-art speech recognition with sequence-to-sequence models," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4774–4778.

[22] ITU-T, *G.114: Transmission Systems and Media, Digital Systems and Networks*, International Telecommunication Union Std., May 2003.

[23] D. Websdale, S. Taylor, and B. Milner, "The effect of real-time constraints on automatic speech animation," in *Interspeech*, 09 2018, pp. 2479–2483.

[24] N. Sadoughi and C. Busso, "Joint learning of speech-driven facial motion with bidirectional long-short term memory," in *International Conference on Intelligent Virtual Agents*. Springer, 2017, pp. 389–402.

[25] C. Luo, J. Yu, X. Li, and Z. Wang, "Realtime speech-driven facial animation using Gaussian mixture models," in *Proceedings of the Multimedia and Expo Workshop (ICMEW)*, July 2014, pp. 1–6.

[26] S. P. Deena, "Visual speech synthesis by learning joint probabilistic models of audio and video," Ph.D. dissertation, School of Computing Sciences, The University of Manchester, 2012.

[27] D. Schabus, M. Pucher, and G. Hofer, "Joint audiovisual hidden semi-Markov model-based speech synthesis," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 2, pp. 336–347, April 2014.

[28] A. Thangthai and B. Theobald, "HMM-based visual speech synthesis using dynamic visemes," in *Proceedings of the Conference on Facial Analysis, Animation and Auditory-Visual Speech Processing (FAAVSP)*, 2015, pp. 88–92.

[29] O. Watts, G. Henter, T. Merritt, Z. Wu, and S. King, "From HMMs to DNNs: where do the improvements come from?" in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2016, pp. 5505–5509.

[30] P. Filntisis, A. Katsamanis, P. Tsiakoulis, and P. Maragos, "Video-realistic expressive audio-visual speech synthesis for the Greek language," *Speech Communication*, vol. 95, pp. 137–152, 2017.

[31] J. Parker, R. Maia, Y. Stylianou, and R. Cipolla, "Expressive visual text to speech and expression adaptation using deep neural networks," in *ICASSP*, 2017, pp. 4920–4924.

[32] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions Signal Processing*, vol. 45, pp. 2673–2681, 1997.

[33] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[34] B. Fan, L. Wang, F. K. Soong, and L. Xie, "Photo-real talking head with deep bidirectional lstm," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4884–4888.

[35] C. Ding, P. Zhu, and L. Xie, "Blstm neural networks for speech driven head motion synthesis," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[36] K. Haag and H. Shimodaira, "Bidirectional lstm networks employing stacked bottleneck features for expressive speech-driven head motion synthesis," in *Intelligent Virtual Agents*, D. Traum, W. Swartout, P. Khooshabeh, S. Kopp, S. Scherer, and A. Leuski, Eds. Cham: Springer International Publishing, 2016, pp. 198–207.

[37] Y. Zhou, Z. Xu, C. Landreth, E. Kalogerakis, S. Maji, and K. Singh, "Visemenet: Audio-driven animator-centric speech animation," *ACM Trans. Graph.*, vol. 37, no. 4, Jul. 2018.

[38] P. Edwards, C. Landreth, E. Fiume, and K. Singh, "Jali: An animator-centric viseme model for expressive lip synchronization," *ACM Trans. Graph.*, vol. 35, no. 4, Jul. 2016.

[39] S. E. Eskimez, R. K. Maddox, C. Xu, and Z. Duan, "Noise-resilient training method for face landmark generation from speech," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 27–38, 2020.

[40] H. X. Pham, Y. Wang, and V. Pavlovic, "End-to-end learning for 3D facial animation from speech," in *Proceedings of the 20th ACM International Conference on Multimodal Interaction*, 2018, p. 361365.

[41] K. Vougioukas, S. Petridis, and M. Pantic, "Realistic speech-driven facial animation with GANs," *International Journal of Computer Vision*, pp. 1–16, 2019.

[42] N. Sadoughi and C. Busso, "Speech-driven expressive talking lips with conditional sequential generative adversarial networks," *IEEE Transactions on Affective Computing*, 2019.

[43] L. Chen, R. K. Maddox, Z. Duan, and C. Xu, "Hierarchical cross-modal talking face generation with dynamic pixel-wise loss," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 7824–7833.

[44] A. Richard, C. Lea, S. Ma, J. Gall, F. de la Torre, and Y. Sheikh, "Audio- and gaze-driven facial animation of codec avatars," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 41–50.

[45] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 681–685, 2001.

[46] I. Matthews and S. Baker, "Active appearance models revisited," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 135–164, 2004.

[47] Y. Lan, R. Harvey, B. Theobald, E.-J. Ong, and R. Bowden, "Comparing visual features for lipreading," in *AVSP*, 2009, pp. 102–106.

[48] J. Gower, "Generalized procrustes analysis," *Psychometrika*, vol. 40, pp. 33–51, 1975.

[49] ETSI, "Speech Processing, Transmission and Quality Aspects (STQ); Distributed speech recognition; Extended advanced front-end feature extraction algorithm; Compression algorithms; Back-end speech reconstruction algorithm," ETSI STQ-Aurora DSR Working Group, ES 202 212 version 1.1.1, Nov. 2003.

[50] A. Mohamed, G. Dahl, and G. Hinton, "Deep belief networks for phone recognition," in *NIPS Workshop Deep Learn. Speech Recognition Related Applicat.*, 2009.

[51] A. Graves, "Generating sequences with recurrent neural networks," *arXiv preprint arXiv:1308.0850*, 2013.

[52] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. ICML*, vol. 30, no. 1, 2013.

[53] N. Harte and E. Gillen, "TCD-TIMIT: An audio-visual corpus of continuous speech," *IEEE Trans. on Multimedia*, vol. 17, no. 5, pp. 603–615, 2015.

[54] L. Wang, Y. J. Wu, X. Zhuang, and F. K. Soong, "Synthesizing visual speech trajectory with minimum generation error," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2011, pp. 4580–4583.