Interspeech 2018
2-6 September 2018, Hyderabad

The Effect of Real-Time Constraints on Automatic Speech Animation

Danny Websdale, Sarah Taylor and Ben Milner

University of East Anglia

d.websdale@uea.ac.uk, s.l.taylor@uea.ac.uk, b.milner@uea.ac.uk

Abstract

Machine learning has previously been applied successfully to speech-driven facial animation. To account for carry-over and anticipatory coarticulation a common approach is to predict the facial pose using a symmetric window of acoustic speech that includes both past and future context. Using future context limits this approach for animating the faces of characters in real-time and networked applications, such as online gaming. An acceptable latency for conversational speech is 200ms and typically network transmission times will consume a significant part of this. Consequently, we consider asymmetric windows by investigating the extent to which decreasing the future context effects the quality of predicted animation using both deep neural networks (DNNs) and bi-directional LSTM recurrent neural networks (BiLSTMs). Specifically we investigate future contexts from 170ms (fully-symmetric) to 0ms (fully-asymmetric). We find that a BiLSTM trained using 70ms of future context is able to predict facial motion of equivalent quality as a DNN trained with 170ms, while introducing increased processing time of only 5ms. Subjective tests using the BiLSTM show that reducing the future context from 170ms to 50ms does not significantly decrease perceived realism. Below 50ms, the perceived realism begins to deteriorate, generating a trade-off between realism and latency.

Index Terms: Real-time speech animation, automatic lip sync.

1. Introduction

Speech animation (or lip sync) is the process of moving the face of a digital character in sync with speech and is an essential component of animated television shows, movies and computer games. Online multi-player games, such as Call of Duty or Overwatch, are increasingly popular and since the early 2000s, voice over IP technologies have enabled players to speak with one another during game play. However, the facial motion of the players' avatars typically does not match the speech. In 2017, Cloud Imperium Games announced *face over IP* technology for adding life to a character's face, but this is based on real-time facial tracking and retargeting of the player's face using a webcam. Real-time speech-driven automatic facial animation would enrich the gaming experience in the same way without the requirement of a webcam. Additionally, it would enable users to interact in Virtual Reality (VR) environments where the face might be partially occluded, and can be used for animating virtual assistants or for animating the voice during a telephone call allowing hearing impaired people to lip read.

Early automatic speech animation approaches were based on placing specific lip poses at key frames (eg. mid-points of each phoneme), and interpolating the in-between frames. Although some efforts were made to add it as a post-process [1], these techniques typically failed to model coarticulation, resulting in low quality animation. Coarticulation is the influence of neighbouring speech on the articulation of a sound, both visually and acoustically. At a particular time t, the articulator position is a complex interaction between the target sound at t, the events leading up to t (carry-over coarticulation), and those following t (anticipatory coarticulation). It is essential that both coarticulatory effects are modelled for realistic speech animation. The challenge addressed in this work lies in how to incorporate anticipatory coarticulation in real-time animation frameworks.

An early real-time solution [2] sampled visual parameters for given acoustic parameters from a joint audio-visual probability distribution, but did not account for coarticulation. More recent solutions have been based on deep learning, which is well suited to this problem due to its ability to learn complex relationships. Networks are commonly trained to estimate the facial pose from a window of speech that contains both past and future acoustic context using either sliding windows [3, 4, 5, 6, 7, 8, 9], or by introducing a time delay [10]. Thus, to output the facial configuration at time t, a window of audio speech surrounding this is required. The rationale behind including past and future speech is to model the effect of both carry-over and anticipatory coarticulation. However, including future context limits many of these approaches to non-real-time graphics applications.

For close to real-time performance, Hong et al. [5] used a 200ms window of audio features centred at time t, introducing a latency of 100ms. Pham et al. [11] computed input speech parameters over fully-asymmetric windows spanning 96ms of past context and no future context. They commented that their model performed *reasonably*, which might be due to its inability to learn anticipatory coarticulation. Suwajanakorn et al. [10] discovered that, given a temporal delay of less than 200ms, a unidirectional LSTM generated poor quality lip sync. Karras et al. [7] observed that they were able to use a future context of 100ms with minor degradation to animation quality. The goal of our work is to look deeper into this effect. Specifically we set out to answer the following questions:

- How much future context is needed for high quality speech-driven facial animation?
- Does the chosen model architecture effect how much future context is needed?
- What trade-offs are required through varying future contexts in terms of the overall latency and the resulting realism?

We experiment with deep neural networks (DNNs) and bi-directional long short term memory models (BiLSTMs) to minimise the amount of future speech context necessary for predicting realistic facial motion. We evaluate both objectively and subjectively and discuss the trade-off between animation quality and processing time.

2. Real-time Considerations

Automatic speech animation in film and television can be created off-line and has no real-time constraints. This allows broad windows of acoustic features to be included when predicting the facial motion. However, for applications that require real-time

10.21437/Interspeech.2018-2066

animation, minimising delay is crucial and this imposes constraints on the amount of future audio context that can be used. A number of studies have investigated the effect of delay in real-time applications, such as the ITU G.114 recommendations on transmission quality for telephony which concludes that delays up to 200ms leave users ‘very satisfied’ while delays from 200-300ms leave users ‘satisfied’ [12].

When animating a character’s face in real-time for online gaming, the time taken from capturing the audio of the player (source) to generating the animated facial movements on the remote players’ (destination) devices is the sum of a number of delays, which broadly comprises future audio look-ahead, network latency (source–server–destination), processing time for speech animation prediction and rendering time.

The animation prediction processing time depends on factors such as the number of audio frames, the algorithm and processor, and as a guide, the baseline system in Section 4.1, requires 0.5ms. Rendering time also depends on various factors, such as scene complexity and available hardware and introduces delay less than the frame rate. Network latency is dependent on the network infrastructure and the distances between the source, server and destination, with a simple estimate being to assume 10μs per mile. For intra-continental routes delays may be up to 30ms, while for inter-continental routes, delays can increase to around 60ms. Additional factors such as the amount of traffic and route taken, introduces variation, or jitter, that can add a further 50% to the network delay.

Ideally, the sum of these delays should be sufficiently small so that users remain very satisfied with the latency of the system (no longer than 200ms). This is unlikely to be achievable with existing methods that require future contexts of, for example, 200ms [10], which leaves insufficient time for processing and network latency. This therefore leads to a trade-off between the overall latency and the realism of the animation, both of which are dependent on the amount of future context.

3. Feature Extraction

Automatic speech animation is a mapping from audio to visual speech, or their respective parameterisations. An effective visual parameterisation must encode the speaker’s facial pose, while the audio parameters must contain information regarding the acoustic speech content.

3.1. Visual Parameterisation

The visual feature extraction used in this work follows that of [13]. We begin with a set of 34 2D vertices that define a mesh demarcating the contours of the lips, jaw and nostrils. An active appearance model (AAM) [14, 15] is used to track and parameterise this facial region, generating a compact 47 dimensional vector, $\mathbf{y}_t$, that encodes both the position and appearance of the jaw area in each frame of the video.

3.2. Acoustic Parameterisation

Given their success in many audio-visual systems, mel frequency cepstral coefficients (MFCCs) are used in this work and follow the method in [6]. These are computed from 20ms frames of speech taken every 10ms. A Hamming window is applied to the frame and the power spectrum computed. A 40 channel mel-filterbank, log operation and discrete cosine transform are subsequently applied before truncating to the first 25 coefficients to give the acoustic feature vectors, $\mathbf{x}_t$.

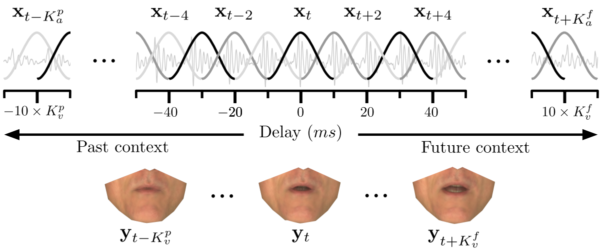


Figure 1: *Audio and visual feature alignment and windowing. The 20ms hamming windowed audio segments with 10ms shift that are used to calculate MFCCs are shown.*

3.3. Audio-visual Alignment and Windowing

Input acoustic windows, $\mathbf{X} = [\mathbf{x}_{t-K_a^p}, \ldots, \mathbf{x}_{t+K_a^f}]$, are formed from MFCC vectors with past contexts of K_a^p vectors and future contexts of K_a^f vectors. These are mapped to output visual windows, $\mathbf{Y} = [\mathbf{y}_{t-K_v^p}, \ldots, \mathbf{y}_{t+K_v^f}]$, with K_v^p and K_v^f past and future visual feature vectors. $\mathbf{X}$ and $\mathbf{Y}$ are aligned at t as illustrated in Figure 1.

Most implementations consider only fully symmetric acoustic windows, such that K_a^p=K_a^f. For real-time applications, the amount of future context (K_a^f) affects the input delay required for model prediction, whereas past context (K_a^p) has no impact (other than processing time). Therefore, we explore asymmetric windows by reducing K_a^f while keeping K_a^p fixed.

4. Model Architectures

We propose using machine learning techniques to predict facial motion given only acoustic speech for generating realistic animation. Specifically, we implement a feed-forward deep neural network (DNN) and compare this with a bi-directional long short-term memory model (BiLSTM) for learning speech related facial motions. Each model is trained to learn the mapping from a window of acoustic features, $\mathbf{X}$, to a window of visual features, $\mathbf{Y}$.

4.1. Feed-forward DNN

As a baseline we use the sliding window DNN approach described in [6]. This system comprises a feed-forward model consisting of 3 hidden dense layers with 2000 rectified linear units (ReLU) [16], and linear output layer. The DNN takes a stacked full window of acoustic features as input $\hat{\mathbf{X}}_t = [\mathbf{x}_{t-K_a^p}; ...; \mathbf{x}_{t+K_a^f}]$ (with stacking function ‘;’), and predicts a stacked output $\hat{\mathbf{Y}}_t = [\mathbf{y}_{t-K_v^p}; ...; \mathbf{y}_{t+K_v^f}]$.

Smooth animation trajectories are generated by first reshaping $\hat{\mathbf{Y}}_t$ into a $\omega_v \times c$ matrix, $\tilde{\mathbf{Y}}_t$, where c are the number of visual coefficients and $\omega_v = (K_v^p + 1 + K_v^f)$, the visual window width, before overlapping and averaging the predicted subsequences:

$$\check{\mathbf{Y}}_t = \frac{1}{\omega_v} \sum_{k=0}^{\omega_v - 1} \tilde{\mathbf{y}}_{(t-K_v^p+k),(t+K_v^f-k)} \qquad (1)$$

where $\tilde{\mathbf{y}}_{t,j}$ is the j^{th} column from the prediction $\tilde{\mathbf{Y}}_t$, at time t. An output comprising only the central visual vector (i.e. $K_v^p = K_v^f = 0$) was considered in preliminary tests but this underperfomed a stacked and smoothed output.

4.2. Bi-directional LSTM

We propose extending the DNN implementation by using bi-directional LSTM models (BiLSTM), which have shown improvements in other speech processing fields such as recognition [17, 18] and TTS synthesis [19], and model the temporal structure found within speech. BiLSTMs process context windows in both forward and backward directions, enabling the model to learn both carry-over and anticipatory coarticulation. This network comprises 2 pairs of recurrent forward and backward layers, each consisting of 256 LSTM cells with peephole connections [20] such that pair $\mathbf{h}^n = [\overrightarrow{\mathbf{h}}_n; \overleftarrow{\mathbf{h}}_n]$, followed by a single 2000 ReLU dense layer and a linear output layer.

The first LSTM pair, $\mathbf{h}^1$, traverses the full context window:

$$\overrightarrow{\mathbf{h}}_1, \overleftarrow{\mathbf{h}}_1 = [\mathbf{x}_{t-K_a^p}, ..., \mathbf{x}_{t+K_a^f}] \qquad (2)$$

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

$$\overrightarrow{\mathbf{h}}_2 = [\mathbf{h}^1_{t-K_a^p}, ..., \mathbf{h}^1_t], \quad \overleftarrow{\mathbf{h}}_2 = [\mathbf{h}^1_t, ..., \mathbf{h}^1_{t+K_a^f}] \qquad (3)$$

Only the final output from $\mathbf{h}^2$, representative of the audio-visual alignment at t, is passed through the remaining network. The network output is equivalent to that of the DNN, and the same overlap and average steps are applied as a post-process (Eq. 1).

5. Experimental Set-up

A set of experiments are described which investigate the effect of future context on accuracy of the predicted visual features, and on perceived realism of the resulting speech animation. The KB-2k audiovisual speech dataset [13] was used for experimentation and comprises an American male uttering ≈2100 phonetically balanced sentences in a neutral style. The audio is sampled at 48kHz while the video is captured at 29.97fps. Sentences were randomly split into training, validation and test sets containing 1884, 100 and 100 sentences respectively.

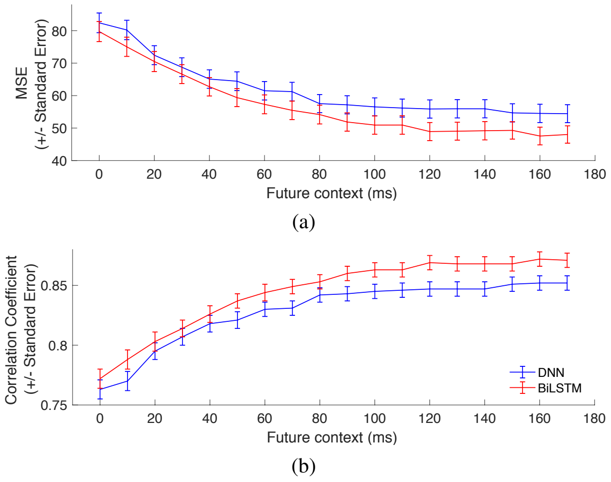
5.1. Model Training

Both the DNN and BiLSTM outlined in Section 4 were implemented within the Lasagne framework [21] with the Theano [22] back-end. Input data was z-score normalised and grouped into minibatches of 256. To prevent overfitting, dropout of 0.5 was added between all layers across both models and early stopping [23] was used when the validation score did not improve after 5 epochs. To speed convergence, batch normalisation was used for all dense layers [24]. Training used backpropagation with the Adam optimiser [25] with a learning rate of 0.0001 and 0.001 for the DNN and BiLSTM respectively, minimising the mean-squared error (MSE) loss function.

Using a 16 core 2.6GHz Tesla K40 GPU, the DNN takes 0.5ms to make 100 predictions given a window of 340ms of acoustic speech, whereas the BiLSTM takes 5ms. Shorter windows require less processing time to produce a prediction.

5.2. Objective Tests

We first examine the effect of future context on the accuracy of the predicted visual features. The study in [6] found that a symmetric acoustic context window of 340ms ($K_a^p = 16$, $K_a^f = 16$ MFCC vectors) produced high quality animation, providing the initial window size for our tests. Our experiments use a fixed past context of 170ms ($K_a^p = 16$ MFCC vectors) while future contexts are varied from 0ms to 170ms ($K_a^f = -1$ to 16 MFCC vectors). For each combination of K_a^p and K_a^f, both DNN and BiLSTM models are trained, and the mean square error (MSE) and correlation coefficient (CC) of the predicted visual parameter vectors for the test set are measured and shown in Figure 2 along with standard error bars. These show that as future context is reduced, the resulting visual feature predictions deteriorate, albeit slowly at first. Reducing future context from 170ms to around 80ms has a small effect, while the impact of further reductions is more severe. The results also show that, for a given future context, the BiLSTM is substantially more effective in predicting visual features than the DNN. This result can also be considered in terms of reducing future context delay. The BiLSTM with just 70ms of future context is able to match the performance of the DNN with 170ms of future context, thereby reducing the latency by 100ms.



(a)

(b)

Figure 2: *a) Mean squared error and b) correlation of predicted and ground truth visual features for increasing future contexts.*

5.3. Subjective Experiments

While objective measures provide an indication of performance, they are not necessarily consistent with the perceived realism of the animation. A subject-based experiment is therefore also performed to determine the effect that varying future context has on the realism of the predicted facial motion. We randomly select 10 sentences from the test partition of the dataset for experimentation.

Since the BiLSTM significantly outperformed the DNN in objective experiments, subjective tests compare only BiLSTM animations using future contexts of $K_a^f = 2, 4, 7, 10, 16$ which is equivalent to durations of 30, 50, 80, 110 and 170ms. All 10 test sentences were rendered in each of the 5 conditions and also with the ground truth (GT) visual features as a baseline, generating 60 clips in total. Each clip was presented in a randomised order to participants who were asked 'Do you think this is real or animated lip motion?'. The participants were offered the options 'Real' or 'Animated'. The experiment was performed using a web interface and participants were recruited by sharing the URL on social media. The first five responses from each of the 79 participants were ignored to allow for familiarisation with the task, leaving an average of 62 votes per clip.

Figure 3 shows the mean perceived realism (MPR) for each future context (red) and for ground truth (black) with one stan-

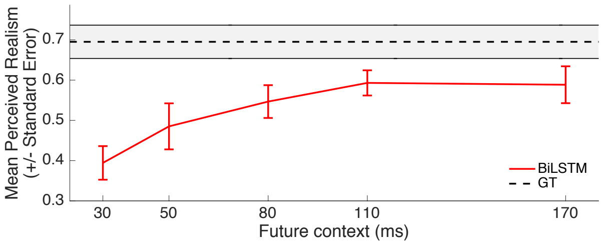


Figure 3: *Mean perceived realism (MPR) of animated sentences generated from a BiLSTM with various future contexts. MPR of ground truth (GT) animations shown as reference.*

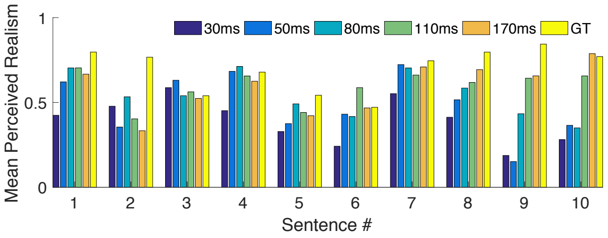


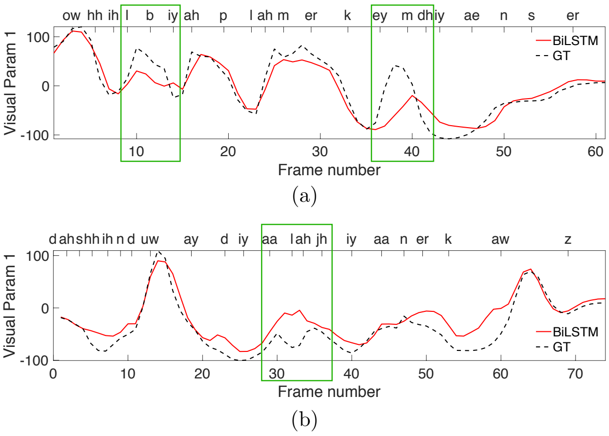
Figure 4: *MPR of each sentence and condition.*

dard error from the mean (error bars and grey region). The MPR is the mean realism rating over all participants and test sentences, and is computed as the number of 'Real' votes divided by the total number of votes for each condition. We observe that only 70% of the ground truth animations are considered real which we attribute to rendering artefacts (eg. inner mouth blurring) introduced by reconstructing images from the visual features. Additionally, the rendered jaw was not composited onto a full face image, rather it was presented in isolation, which may also have been distracting to participants.

As would be expected, highest MPR is achieved with the largest future context of 170ms which was perceived as real 59% of the time — 11% lower than the ground truth condition. Shortening the future context to 110ms has no effect on realism but values less than this do cause a reduction. With 30ms lookahead the animation is only perceived as realistic 39% of the time. A paired t-test finds that increasing the future context from 30ms to 50ms gives a significant increase in MPR ($p = 0.03$). Extending the context to 80, 110 and 170ms produces a considerable, but not significant increase in MPR compared to the 50ms responses ($p = 0.11, 0.08, 0.15$ respectively).

Figure 4 shows the MPR per test sentence. Clearly, the perceived realism is dependent on the speech content of the sentence. The model generates animation with MPR very close to ground truth in most cases, but performs poorly on sentence 2. The first predicted visual parameter for sentence 2 with 170ms of future context can be seen in Figure 5a. This sentence corresponds to the audio "Oh he'll be a plumber, came the answer". In contrast, Figure 5b shows the first visual parameter for sentence 3 with just 30ms, which has high MPR, and corresponds to the speech "Does Hindu ideology honour cows?".

Both predicted trajectories follow closely the curve of the ground truth trajectories, but fail to reach targets in certain regions highlighted in green. Sentence 2 contains multiple frontal consonants to which viewers are highly sensitive, such as the bilabial consonants in the words *be* (frame 12) and *came* (frame 39). We observe under-articulated lip closures at these frames, which could have made the entire sentence appear implausible.



(a)

(b)

Figure 5: *First dimension of predicted (red) and ground truth (black) visual parameters generated using a BiLSTM for (a) test sentence 2 with 170ms of future context and (b) sentence 3 with 30ms. The corresponding phoneme labels are shown above.*

Sentence 3 contains fewer frontal sounds, making it less susceptible to this effect. We expect the MPR to vary across sentences since the forward reach of anticipatory coarticulation has been found to be dependent on the speech context, and can extend beyond 200ms in certain cases [26].

6. Conclusion

The aim of this work has been to explore the trade-off between animation realism and overall latency within a real-time application such as online gaming. The quality of animation has been shown to be related to both the model architecture and the future acoustic context from which visual feature predictions are made. The BiLSTM was found to be significantly better than a DNN, the benefit of which can be used either to reduce errors in prediction or to allow future context to be reduced. Analysis looking specifically at the effect of future context showed that a reduction from 170ms to 110ms has no effect on realism, while reducing to 80ms has a small effect.

Given that the dominant delays with online gaming applications are the future context and network latency it is required that these together should be no more than the desired overall delay. The ITU G.114 recommendation indicates no reduction in user satisfaction for delays below 200ms. Taking this as a target delay, and assuming a worst case delay of 100ms for network transmission and prediction processing time, this allows 100ms of future context which has been shown to cause no perceptual reduction in realism. In situations where network latencies are longer, the same overall latency can be achieved by reducing future context although it is likely that realism will degrade. Similarly, if a shorter overall delay is required, again future context can be reduced albeit with a degradation of realism. This highlights the inter-relationship between realism and overall latency, both of which are affected by the future context. The specific choice of the future context is therefore dependent on factors such as the network latency, the overall latency target and the level of realism required in the animation.

7. Acknowledgements

This work was supported by UK EPSRC award EP/M014053/1, UEA's HPC and by NVIDIA.

8. References

[1] M. M. Cohen and D. W. Massaro, "Modeling coarticulation in synthetic visual speech," in *Models and techniques in computer animation*. Springer, 1993, pp. 139–156.

[2] R. Rao, T. Chen, and R. Mersereau, "Exploiting audio-visual correlation in coding of talking head sequences," in *International Picture Coding Symposium*, vol. 28, 1996.

[3] D. W. Massaro, J. Beskow, M. M. Cohen, C. L. Fry, and T. Rodgriguez, "Picture my voice: Audio to visual speech synthesis using artificial neural networks," in *AVSP'99-International Conference on Auditory-Visual Speech Processing*, 1999.

[4] G. Takács, "Direct, modular and hybrid audio to visual speech conversion methods-a comparative study." in *Interspeech*, 2009, pp. 2267–2270.

[5] P. Hong, Z. Wen, and T. S. Huang, "Real-time speech-driven face animation with expressions using neural networks," *IEEE Transactions on Neural Networks*, vol. 13, no. 4, pp. 916–927, Jul 2002.

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

[7] 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 Transactions on Graphics (TOG)*, vol. 36, no. 4, p. 94, 2017.

[8] 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*, 2017, pp. 389–402.

[9] B. Fan, L. Wang, F. Soong, and L. Xie, "Photo-real talking head with deep bidirectional lstm," in *International Conference on Acoustics, Speech and Signal Processing*, 2015.

[10] S. Suwajanakorn, S. M. Seitz, and I. Kemelmacher-Shlizerman, "Synthesizing obama: learning lip sync from audio," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, p. 95, 2017.

[11] H. X. Pham, Y. Wang, and V. Pavlovic, "End-to-end learning for 3d facial animation from raw waveforms of speech," *CoRR*, vol. abs/1710.00920, 2017.

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

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

[14] 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.

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

[16] 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.

[17] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural Networks*, vol. 18, no. 5-6, pp. 602–610, 2005.

[18] A. Graves, N. Jaitly, and A.-r. Mohamed, "Hybrid speech recognition with deep bidirectional lstm," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*. IEEE, 2013, pp. 273–278.

[19] Y. Fan, Y. Qian, F.-L. Xie, and F. K. Soong, "Tts synthesis with bidirectional lstm based recurrent neural networks," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

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

[21] S. Dieleman, J. Schlter, C. Raffel, E. Olson, S. K. Snderby, D. Nouri *et al.*, "Lasagne: First release." Aug. 2015. [Online]. Available: http://dx.doi.org/10.5281/zenodo.27878

[22] Theano Development Team, "Theano: A Python framework for fast computation of mathematical expressions," *arXiv e-prints*, vol. abs/1605.02688, May 2016. [Online]. Available: http://arxiv.org/abs/1605.02688

[23] L. Prechelt, "Early stopping-but when?" in *Neural Networks: Tricks of the trade*. Springer, 1998, pp. 55–69.

[24] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, 2015, pp. 448–456.

[25] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[26] J.-L. Schwartz and C. 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*, vol. 10, no. 7, pp. 1–10, 07 2014.