Speech-driven 3D Facial Animation with Implicit Emotional Awareness: A Deep Learning Approach

Hai X. Pham¹, Samuel Cheung², and Vladimir Pavlovic¹

¹Department of Computer Science, Rutgers University
{hxp1,vladimir}@cs.rutgers.edu
²Department of Electrical and Computer Engineering, Rutgers University
sc1528@rutgers.edu

Abstract

We introduce a long short-term memory recurrent neural network (LSTM-RNN) approach for real-time facial animation, which automatically estimates head rotation and facial action unit activations of a speaker from just her speech. Specifically, the time-varying contextual non-linear mapping between audio stream and visual facial movements is realized by training a LSTM neural network on a large audio-visual data corpus. In this work, we extract a set of acoustic features from input audio, including Mel-scaled spectrogram, Mel frequency cepstral coefficients and chromagram that can effectively represent both contextual progression and emotional intensity of the speech. Output facial movements are characterized by 3D rotation and blending expression weights of a blendshape model, which can be used directly for animation. Thus, even though our model does not explicitly predict the affective states of the target speaker, her emotional manifestation is recreated via expression weights of the face model. Experiments on an evaluation dataset of different speakers across a wide range of affective states demonstrate promising results of our approach in real-time speech-driven facial animation.

1. Introduction

Human-machine interaction has been one active research area for decades, with the ultimate goal to make human-machine interaction transparent. Speech, as a natural form of communication among various modes of interactions, is becoming more immersive, evidenced by the increasing popularity of virtual voice assistants, such as Microsoft’s Cortana or Amazon’s Alexa, in our daily lives. Furthermore, not only the contextual sound units (phonemes) are carried in the audio recording, but also emotional states of the speaker via speed or intensity of her speech [18, 6, 4, 17, 5]. Thus, a lively animated 3D head representing the speaker will certainly enhance the speech perception experience in many applications. One such application is the development of talking agent, either in the form of virtual or physical (i.e. robotic) avatars, for face-to-face human-machine interaction, as in computer-assisted voice agent. In this scenario, the recorded speech can easily be manipulated, by changing the speed or pitch, to reflect the artificial emotion of the digital assistant. These changes can be automatically reflected visually on the avatar, and make the interaction more engaging. On the other hand, it can also make inter-person telecommunication more enjoyable by expressing speech via personalized avatars, especially in interactive role-playing games, where the gamers communicate with other characters in the virtual world.

In this work, we aim to recreate a talking 3D virtual avatar that can naturally rotate and make micro facial movements to reflect the time-varying contextual information and emotional intensity carried in the input speech. Intuitively, this work is analogous to visual 3D face tracking [20, 19], however, it is more challenging as we try to map acoustic sequence to visual space, instead of conveniently relying on textural cues from input images. Moreover, speech-emitting facial movements involve different activations of correlated regions on the geometric surface, thus it is difficult to achieve realistic looking, emotion-aware facial deformation from speech sequence.

Thus, we propose a regression framework based on long short-term memory recurrent neural network to estimate rotation and activation parameters of a 3D blendshape face model [7] from sequence of acoustic features, for real-time life-like facial animation. We extract a wide range of acoustics features to capture contextual and emotional progression of the speech. To tackle the difficulty of avatar gen-
eration, we utilize the blendshape model in [7], which is purposely designed with enough constraints to ensure that, the final model would always look realistic given a specific set of control parameters. In addition, it can represent various emotional states, e.g. sadness, happiness, etc., without explicitly specifying them. In order to directly map the input features to face shape parameters, we use deep recurrent neural network with LSTM cells [15] to model the long range context of the sequence.

2. Related Work

Text or speech-driven facial animation. Usually related in the literature as "talking head", various approaches have been developed to animate a face model driven by either text or speech. A text-driven approach typically consists of a text-to-speech and a text-to-face shape synthesizing unit, and are combined to generate facial animation [24, 9]. Speech-driven techniques often share a common approach: directly map an input sequence of acoustic features to a sequence of visual features [13, 29, 22].

The above approaches can also be categorized according to the underlying face model, into model-based [2, 1, 23, 28, 10, 8] and image-based [3, 9, 12, 25, 29, 13]. Image-based approaches compose the output video by concatenating short clips, or stitch different regions from a sample database identified by a classifier, together. These approaches usually generate photo-realistic video output, as they compose the result from real images with natural textures. However, their performance and quality are limited by the amount of samples in the database, thus it is difficult to generalize to a large corpus of speeches, which would require a tremendous amount of image samples to cover all possible facial appearances. In contrast, although lacking in photo-realism, model-based approaches enjoy the flexibility of a deformable model, which is controlled by only a set of parameters, and more straightforward modeling.

Essentially, every talking head animation technique requires a particular algorithm in order to map an input to visual features, which can be formulated as a regression or classification task. Classification approaches usually identify phonetic units (phonemes) from speech and map to visual units (visemes) based on specific rules, and animation is generated by morphing these key images. Regression approaches, on the other hand, can directly estimate visual parameters from input features and generate continuous trajectories. Early successes in speech-driven talking head were achieved by using Hidden Markov models (HMMs) for trajectory estimation [25, 26]. However, HMM-based techniques incur certain limitations of generative model, e.g. wrong model assumption, or over-smoothing because of the maximum likelihood framework. In recent years, deep neural networks have been successfully applied to speech synthesis [21, 30] and facial animation [10, 8, 11, 13] with superior performance. This is because deep neural networks are able to learn the correlation of high-dimensional input data, and, in case of recurrent neural network, long-term relation, as well as the highly non-linear mapping between input and output features.

Long short-term memory recurrent neural networks. Recurrent neural networks (RNNs) [27] have demonstrated highly desirable performance in sequence modeling with the ability to integrate temporal contextual information. Hochreiter et al. [15] introduced the Long short-term memory (LSTM) cell in RNN to overcome the vanishing gradient problem [14] in modeling long-term relation. In this work, we aim to estimate the facial transformation trajectory in real-time, hence we utilize unidirectional (forward) LSTM-RNN that only memorizes the past data.

3. System Overview

Figure 1 illustrates the architecture of our proposed speech-driven facial animation framework, which includes a training stage and an animation stage. In the training phase, the speech-to-facial parameters mapping is learned by a LSTM-RNN model from the RAVDESS database [16], a large audio-visual corpus that consists of high resolution videos of various speeches and emotions (cf. 6 for more details). High quality videos allow accurate visual tracking of 3D facial deformations, which subsequently enable the deep model to learn complex mapping between speeches and facial actions. Our method is totally language-independent, hence it can be extended with more samples of other subjects speaking different languages. In the animation phase, the trained LSTM model converts input acoustic features into head rotation and facial deformation parameters to drive a 3D blendshape face model.

At the first step in the training phase, various input acoustic features and expected visual output, including head rotation and local deformation parameters, are extracted from training videos (cf. 6 for details on feature and parameter extraction). Subsequently, they are used to train a discriminative LSTM-RNN model by minimizing a squared loss, in order to effectively learn the non-linear mapping between input features and output parameters. The animation phase is very straightforward: given a recorded speech sequence and its features, the LSTM-RNN model estimates head rotation and deformation parameters, which are then used to animate a 3D face model to visually recreate the facial movements and expression carried in the input speech.

4. Feature Representation

4.1. Face Model Parameterization

In this work, we utilize the 3D blendshape face model from the FaceWarehouse database [7], in which, an arbitrary shape $S$ including head pose of a subject can be com-
posed as:

\[ S = R \left( B_0 + \sum_{i=1}^{N} (B_i - B_0)e_i \right), \]  \hspace{1cm} (1)

where \((R, e)\) are rotation and expression blending parameters, respectively, \(\{B_i| i = 1...N\}\) are personalized expression blendshape bases of a particular person, calculated from a 3D tensor, and their combinations are consistent across users. Note that \(\{e_i\}\) are constrained within \([0, 1]\). Readers are encouraged to find more details about the Face-Warehouse face model and its parameterization in [7].

Rotation and deformation parameters \((R, e)\) are the output of our deep model, where \(R\) is represented by three free parameters of a quaternion. In our implementation, the number of expression bases \(N\) is 46, hence the output parameter vector holds 49 values in total. We use the real-time 3D face tracker in [20], using only RGB input, to extract these parameters from training videos. In particular, the face tracker recovers facial parameters in each input video frame by performing two steps: 3D face alignment and refinement. In the alignment step, 3D facial parameters are rapidly estimated by a random forest-based heterogeneous regression pipeline trained upon regular image datasets, which also predicts 2D landmarks corresponding to a set of specific 3D vertices of the blendshape model in order to account for unseen identities and expressions. In such cases, 2D displacement errors tend to be large, i.e. the predicted 2D landmarks differ from the 2D projection of their corresponding 3D vertices considerably, and these errors are minimized in the subsequent refinement step. In this step, 3D facial parameters are fine-tuned by deforming the 3D face model to fit 2D landmarks estimated by the regressor, while maintaining temporal coherency w.r.t. previous frames. Figure 2 shows a few sample frames from the RAVDESS training set.

Figure 2: A few samples from the training data, where a 3D facial blendshape is aligned to the face of the actor in the input frame. Green dots mark 3D landmarks of the model projected to image plane. The blendshape rendered here is, however, a generic model animated given parameters estimated by the tracker. We also use this 3D shape model in our animation experiments.

4.2. Input Feature Extraction

The input to our system can be any arbitrary speech of any length. As we only use low-level acoustic features, our model is not tied to any particular language, and it can be easily extended given more training samples. Specifically, we extract Mel-scaled spectrogram, Mel frequency cepstral coefficients (MFCCs) and chromagram from the audio sequence. Mel-scaled spectrogram and MFCCs are standard
acoustic features proven to be very effective in presenting the contextual information, whereas chromagram is necessary to determine the pitch in the speech, which reflects the affective states of the speaker throughout the entire sequence.

We assume that every input audio sequence is synchronized to the corresponding video at 30 FPS and the audio sampling rate is at 44.1 kHz. Thus, for every video frame, there are 1,470 corresponding audio samples. We include additional samples from the previous video frame, such that for each video frame there is enough audio data to extract three windows of 25ms each, with hop length of 512 samples. In every audio window, values of 128 Mel bands, 13 Mel frequency cepstral coefficients and their delta and delta-delta coefficients, and 12 chroma bins, are extracted. In summary, the input feature vector for every video frame has 537 dimensions, and each variable is normalized to zero mean - unit variance. Figure 4 illustrates different feature sequences extracted from videos of the same actor speaking the same sentence in different emotional states.

5. Deep LSTM-RNN for Facial Animation

5.1. LSTM-RNN

Recurrent neural networks (RNNs) have the ability to memorize past inputs in internal states. They are able to incorporate temporal contextual information, thus RNNs are very suitable for sequence modeling. However, conventional RNNs can only remember limited range of past context because of the vanishing gradient problem [14]. Long short-term memory (LSTM) unit, shown in Figure 5, is designed to overcome this limitation. LSTM unit is able to store its value for long period of time by controlling the flow of information into and out of its memory. A forward pass in the recurrent hidden layer of LSTM-RNN is as follows:

\[
\begin{align*}
    i_t & = \sigma (W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i), \\
    f_t & = \sigma (W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f), \\
    o_t & = \tau (W_{xo}x_t + W_{ho}h_{t-1} + b_o), \\
    c_t & = f_tC_{t-1} + i_tC_i, \\
    h_t & = o_Tc_t, \\
    y_t & = \eta (W_{hy}h_t + b_y),
\end{align*}
\]

where σ and τ are sigmoid and tanh activation functions, i, o, f, a and c are input gate, output gate, forget gate, cell input activation and cell memory, respectively. \( t = 1..T \), where \( T \) is the sequence length. \( x_t \) is the input at time \( t \), while \( h_t \) is the output of the hidden layer, \( y_t \) is the final output of the network, and \( \eta \) is the activation function of the output layer. \( \{W\} \) and \( \{b\} \) are weight matrices and bias vectors, respectively.

5.2. LSTM-RNN for Facial Action and Rotation Synthesis

Our framework maps input acoustic feature vector sequence of \( x_t, t = 1..T \) to output sequence of shape parameter vectors \( y_t \), where \( T \) is the number of video frames. Thus, at any given time \( t \), the deep LSTM model estimates \( y_t = (R_t, e_t) \) from an input feature vector \( x_t \). Blending
weights $e_t$ in particular have to be constrained within $[0, 1]$, hence we split the output into two separate layers, $Y_R$ for rotation and $Y_e$ for expression weights. $Y_R$ is simply a linear layer, whereas $Y_e$ uses ReLU activation to enforce non-negativity on the output:

$$
Y_{ht} = W_{hyt} h_t + b_{yht},
$$

$$
Y_{et} = \text{ReLU} \left( W_{hye} h_t + b_{yee} \right),
$$

$$
y_t = (Y_{ht}, Y_{et}).
$$

The architecture of our deep LSTM-RNN model is illustrated in Figure 6. We train the model by minimizing the square error:

$$
E = \sum_t \left\| y_t - \hat{y}_t \right\|^2,
$$

where $\hat{y}_t$ is the expected output, which we extract from training videos.

6. Experiments

Dataset. We use the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) [10] for training and evaluation. Specifically, the database consists of 24 professional actors (12 male and 12 female, respectively) speaking and singing with various emotions. The speech set consists of eight general emotional expressions: neutral, calm, happy, sad, angry, fearful, surprise, and disgust, where each video sequence is associated with one among eight affective states. Similarly, the song set, in which the actors sing short sentences, consists of six general emotional expressions: neutral, calm, happy, sad, angry, and fearful. Both sets are used for training and testing. We use video sequences of the first 20 actors for training, with around 250,000 frames in total, and evaluate the model on the data of four remaining actors.

Implementation Details. Our framework [1] is implemented in Python, based on the deep learning toolkit CNTK [2]. We train the deep models in 300 epochs, where learning rate is chosen as 0.003 for the first two epochs, 0.0015 for the next 12, and 0.0003 for the remaining epochs. Excluding the time for acoustic feature extraction, it takes about 5ms on average to estimate output vector $y_t$ from one input frame $x_t$, on a laptop equipped with a relatively low-end Quadro K1000M GPU. Thus, our model is suitable for real-time speech-driven animation task.

Evaluations. We train and evaluate performance of three different LSTM-RNN topologies, as listed in Table 1 in which we also compare their performance with support vector regression (SVR) [11]. The metrics in this table is mean squared error of parameters over all video frames in the held-out test set:

$$
\varepsilon = \frac{1}{M} \sum_{i=1}^{M} \left\| y_i - \hat{y}_i \right\|^2,
$$

where $M$ is the number of video frames, $y$ and $\hat{y}$ are model and expected output, respectively. According to this table, all deep models outperform the SVR baseline, and Net 1 achieves the smallest training error after 300 epochs, but its testing error is slightly higher than other two network models across all affective states in the database. Moreover, we use the speech model-generated parameters and test-specific blendshape expression units estimated by the visual tracker to calculate a person-specific 3D shape as in (1), and extract its landmarks in order to compare to visual tracking results. In particular, Table 2 and 3 contain root mean squared error (RMSE) of 3D landmarks in millimeters and RMSE of projected 2D landmarks normalized over average head size, which equals to 400 pixels in RAVDESS, respectively. Figure 8 shows histogram of landmark errors of a few key landmarks.

According to Table 2 the average 3D error of deep models is about 10mm. It is expected because speech-driven models cannot accurately estimate head rotation. Specifically, each actor has a different person-specific rigid head movement pattern, whereas the deep models learn to estimate head rotation by averaging over all training samples as in (4). Thus, these models are unable to effectively reproduce accurate rigid head movement on the test data. However, rigid rotations, temporally smoothed by RNN, can augment the 3D animation for more realistic visual effect, compared to a stationary head. In general, in terms of shape error, all three network models achieve similar errors, with Net 3 slightly outperforms the other two, but the difference in 3D error is less than 1mm. These results demonstrate that our deep models can generate consistently realistic 3D

---

1Code available at: research.cs.rutgers.edu/~hxpl/SpeechProject.html
2Microsoft Cognitive Toolkit (cntk.ai)
Figure 7: Two sequences from the test set. For each sequence, shown from top to bottom: the original video, the 3D blendshape animated by the visual tracker [20] and the speech-driven animation generated by our deep model, respectively.

Table 1: Training and testing errors of different network configurations. For example, 1:600-2:200 means this is a 2-hidden-layer LSTM-RNN whose 1st and 2nd hidden layers have 600 and 200 units, respectively. Testing error is also separated corresponding to eight affective states, e.g. the "Happy" column contains mean squared error over all frames in videos labeled with "happy" general emotion.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Train. Err.</th>
<th>Test Err.</th>
<th>Neutral</th>
<th>Calm</th>
<th>Happy</th>
<th>Sad</th>
<th>Angry</th>
<th>Fear</th>
<th>Disgust</th>
<th>Surprised</th>
</tr>
</thead>
<tbody>
<tr>
<td>Net 1</td>
<td>1:600-2:200</td>
<td>0.59</td>
<td>3.22</td>
<td>3.22</td>
<td>3.44</td>
<td>3.27</td>
<td>2.96</td>
<td>3.06</td>
<td>3.53</td>
<td>3.17</td>
</tr>
<tr>
<td>Net 2</td>
<td>1:300-2:200</td>
<td>1.08</td>
<td>3.03</td>
<td>2.98</td>
<td>3.25</td>
<td>3.09</td>
<td>2.81</td>
<td>2.89</td>
<td>3.30</td>
<td>3.10</td>
</tr>
<tr>
<td>Net 3</td>
<td>1:600-2:600</td>
<td>0.79</td>
<td>3.06</td>
<td>3.02</td>
<td>3.14</td>
<td>3.10</td>
<td>2.69</td>
<td>2.94</td>
<td>3.30</td>
<td>3.02</td>
</tr>
<tr>
<td>SVR</td>
<td>n/a</td>
<td>3.39</td>
<td>3.50</td>
<td>3.52</td>
<td>3.40</td>
<td>3.84</td>
<td>3.45</td>
<td>3.32</td>
<td>3.34</td>
<td>3.62</td>
</tr>
</tbody>
</table>

Table 1: Training and testing errors of different network configurations. For example, 1:600-2:200 means this is a 2-hidden-layer LSTM-RNN whose 1st and 2nd hidden layers have 600 and 200 units, respectively. Testing error is also separated corresponding to eight affective states, e.g. the "Happy" column contains mean squared error over all frames in videos labeled with "happy" general emotion.
Poorest estimation performance on "happy" sequences in general, as shown in Table 1, can be accounted for by a couple of reasons. First, the amount of "happy" frames is only one seventh of the entire dataset. It causes bias of the model towards more similar emotions, such as "angry", "fearful" and "disgust". Second, it is difficult to distinguish the "happy" speech from "surprised" or "angry" from speech alone as under these emotions, speakers tend to speak equally loudly. Lastly, although smiling is a strong visual cue, it is not reflected via speech in an obvious way. Note that in these "happy" sequences, the actors do not actually laugh at any time, but rather smile while speaking, hence it is difficult to recognize the smiling gesture from speech. More analysis on "happy" speeches is required in order to identify the smiling cue from audio data.

Table 2: RMSE (in mm) when comparing reconstructed 3D landmarks using the speech model-generated parameters to visual tracking results. Net 3 achieves smaller error overall.

<table>
<thead>
<tr>
<th>Overall</th>
<th>Neutral</th>
<th>Calm</th>
<th>Happy</th>
<th>Sad</th>
<th>Angry</th>
<th>Fear</th>
<th>Disgust</th>
<th>Surprised</th>
</tr>
</thead>
<tbody>
<tr>
<td>Net 3</td>
<td>9.91</td>
<td>8.60</td>
<td>9.90</td>
<td>8.96</td>
<td>10.68</td>
<td>8.84</td>
<td>10.60</td>
<td>12.39</td>
</tr>
<tr>
<td>SVR</td>
<td>20.30</td>
<td>21.55</td>
<td>30.33</td>
<td>20.62</td>
<td>19.88</td>
<td>18.82</td>
<td>19.63</td>
<td>23.06</td>
</tr>
</tbody>
</table>

Table 3: Normalized RMSE when comparing reconstructed 2D landmarks using the speech model-generated parameters to visual tracking results. 2D landmarks are created by projecting 3D corresponding landmarks onto the image plane.

<table>
<thead>
<tr>
<th>Overall</th>
<th>Neutral</th>
<th>Calm</th>
<th>Happy</th>
<th>Sad</th>
<th>Angry</th>
<th>Fear</th>
<th>Disgust</th>
<th>Surprised</th>
</tr>
</thead>
<tbody>
<tr>
<td>Net 1</td>
<td>0.050</td>
<td>0.041</td>
<td>0.048</td>
<td>0.045</td>
<td>0.055</td>
<td>0.045</td>
<td>0.053</td>
<td>0.063</td>
</tr>
<tr>
<td>Net 2</td>
<td>0.053</td>
<td>0.045</td>
<td>0.054</td>
<td>0.047</td>
<td>0.056</td>
<td>0.047</td>
<td>0.054</td>
<td>0.070</td>
</tr>
<tr>
<td>Net 3</td>
<td>0.050</td>
<td>0.040</td>
<td>0.050</td>
<td>0.044</td>
<td>0.054</td>
<td>0.044</td>
<td>0.053</td>
<td>0.063</td>
</tr>
<tr>
<td>SVR</td>
<td>0.070</td>
<td>0.064</td>
<td>0.073</td>
<td>0.066</td>
<td>0.073</td>
<td>0.066</td>
<td>0.073</td>
<td>0.079</td>
</tr>
</tbody>
</table>

Facial animation, despite the limitation in rigid motion estimation, thanks to the underlying blendshape model. However, a thorough user study is desirable in order to measure the quality of animation generated by these models, which we look forwards to conducting in the future.

From our visual observation, Net 1 consistently outperforms other networks, in terms of local facial deformation quality, especially in lower lip movements, indicated by smaller landmark error as shown in Figure 8. In order to explain this phenomenon, we further categorize estimation errors into separate bins for individual blending weights $e_i$ for each network, as shown in Figure 9.

According to this figure, Net 2 and Net 3 have very similar errors across all coefficients with the exception of units 20 and 24, whereas Net 1 has higher errors with expression units 7, 8, 32 and 46. However, these units carry very subtle facial deformations, and thus do not affect the face reconstruction quality of Net 1 in general. A possible explanation is that, Net 2 and Net 3 tend to smooth output parameters to achieve lower mean error, trading off the ability to model spontaneity of facial expressions. Further investigation is needed in order to understand how the unbalanced architecture of Net 1 contributes to this phenomenon. Figure 10 demonstrates that Net 1 achieves smallest errors on the parameters most relevant to the "Surprised" state of Actor 24.

Figure 7 shows two example sequences from the test set. Both actors speak the same sentence, "Kids are talking by the door", but under different emotions, tones and speeds. Facial parameters are estimated by Net 1. Speech-driven animation quality on the first sequence from Actor 24 is rather good, where micro facial movements estimated by our model match closely to that of the visual tracker [20]. However, our model cannot effectively recreate lip deformations in the "Happy" sequence of Actor 21.

7. Conclusion and Future Work

This paper presents a deep recurrent learning approach for speech-driven 3D facial animation. Our regression framework, based on deep long short-term memory recurrent neural network, directly maps various acoustic features of an input speech sequence to head rotation and facial deformation parameters of a 3D blendshape model for realistic animation in real-time. Experimental results on a real audio-visual corpus consisting of speeches under various emotions demonstrate the effectiveness of our approach in recreating the affective state and facial deformation of the speaker. We believe our work is a reasonably good baseline for further research in speech-driven facial animation. In the future, we will explore the ability to learn features directly from the raw waveform data, and incorporate deep generative model in our framework to improve its facial parameter generation quality.
Figure 8: Histograms of landmark error distributions corresponding to eyebrows, upper eyelids, chin, cheeks, upper lip and lower lip. Data points in bins closer to the center of the histogram have smaller errors.

Figure 9: Parameters errors of three LSTM models, categorized by expression units.

Figure 10: Parameter errors across time of the "Actor 24 - Surprised" sequence. We plot errors corresponding to six action units: left/right eyebrow raiser small/strong and jaw dropper small/strong. From top to bottom: plots of of Net 1, Net 2, Net 3 and SVR, respectively. Net 1 achieves smallest errors w.r.t. these action unit parameters. Error curves of the deep models are smoother than that of SVR, thanks to temporal coherency imposed by recurrent networks.
References


