Multimed Tools Appl
DOI 10.1007/s11042-015-2944-3

A deep bidirectional LSTM approach for video-realistic talking head

Bo Fan[1] · Lei Xie[1] · Shan Yang[1] · Lijuan Wang[2] · Frank K. Soong[2]

Received: 19 June 2015 / Revised: 2 September 2015 / Accepted: 8 September 2015

Abstract This paper proposes a deep bidirectional long short-term memory approach in modeling the long contextual, nonlinear mapping between audio and visual streams for video-realistic talking head. In training stage, an audio-visual stereo database is firstly recorded as a subject talking to a camera. The audio streams are converted into acoustic feature, i.e. Mel-Frequency Cepstrum Coefficients (MFCCs), and their textual labels are also extracted. The visual streams, in particular, the lower face region, are compactly represented by active appearance model (AAM) parameters by which the shape and texture variations can be jointly modeled. Given pairs of the audio and visual parameter sequence, a DBLSTM model is trained to learn the sequence mapping from audio to visual space. For any unseen speech audio, whether it is original recorded or synthesized by text-to-speech (TTS), the trained DBLSTM model can predict a convincing AAM parameter trajectory for the lower face animation. To further improve the realism of the proposed talking head, the trajectory tiling method is adopted to use the DBLSTM predicted AAM trajectory as a guide to select a smooth real sample image sequence from the recorded database. We then stitch the selected lower face image sequence back to a background face video of the same subject, resulting in a video-realistic talking head. Experimental results show that the proposed DBLSTM approach outperforms the existing HMM-based approach in both objective and subjective evaluations.

Keywords Talking head · Visual speech synthesis · Recurrent neural network · Long short-term memory · Active appearance model

✉ Bo Fan
fanbo1990@gmail.com

Lijuan Wang
lijuanw@microsoft.com

✉ Lei Xie
xielei21st@gmail.com

[1] School of Computer Science, Northwestern Polytechnical University, Xian, China

[2] Microsoft Research Asia, Beijing, China

Published online: 29 September 2015

1 Introduction

Immersive interaction between human and machines has been a popular research area for several decades. Among various ways of interaction, human-machine speech communication is one of the most immersive ways because speech is the primary natural communication means between humans. Speech production and perception are both bimodal in nature. That means, visual speech, i.e., speech-evoked facial movements, plays an indispensable role in speech communication. Therefore, human-machine speech commutation will become more immersive if a vivid talking head is present. Microsoft has recently shown a prototype of talking agent towards human-machine face-to-face interaction [40]. A lively, lip-sync talking head is able to attract the attention of a user, making the human-machine interface more engaging. On the other hand, talking heads are even able to make inter-person tele-communication more interesting and enjoyable. Microsoft has released Avatar Kinect [13], which is an Xbox 360 chatroom-type service that allows you and your friends to talk to each other using personalized talking avatars. Talking heads can be useful in many applications, e.g., reading emails, news or eBooks, playing characters in computer games, acting as an intelligent voice agent or a computer assisted language tutor, etc.

We aim to achieve an *immersive* photo-real talking head where the animation looks video realistic. In other words, we desire our talking head to look as much as possible as if it were a camera recording of a human face, and not that of a cartoon character. Achieving a photo-real talking head is quite challenging. As we know, the human face has an extremely complex geometric form [30] and the speech-originated facial movements are the result of a complicated interaction between a number of anatomical layers that include bone, muscle, fat and skin. Consequently, we are extremely sensitive to the slightest artifacts in an animated talking face and even the most subtle changes can lead to unrealistic appearance.

In this paper, we propose a long short-term memory recurrent neural network (LSTM-RNN) approach for video-realistic talking head. Recently, deep neural network (DNN) based approaches have shown superior performance in many tasks, such as speech recognition [15], speech synthesis [49], natural language processing [4] and computer vision [22]. There are two mainstream neural net structures: feed forward and recurrent. Specifically, recurrent neural networks (RNN) with purpose-build long short-term memory (LSTM) cells can incorporate long range context from the input sequence [17]. They have shown strong trajectory modeling ability in speech recognition [12] and synthesis [10] tasks. Therefore, in this paper, we propose to use LSTM-RNNs to learn a direct mapping between the input speech/text and the output facial movements.

First, the audio/visual parallel training data from a subject are converted into sequences of input and output feature vectors, respectively. The input feature vector can be contextual labels or acoustic representations of speech. Like [47], we adopt the active appearance model (AAM) to model the lower face of the subject and take the low dimensional visual appearance parameters as the output features. After that, an LSTM-RNN is trained to learn the regression model between the input and output sequences by minimizing the generation errors. In the synthesis stage, the LSTM-RNN is used to predict the AAM parameter sequence from a given input sequence. The generated AAM trajectory is then used as a guide to select, from the original training database, an optimal sequence of facial images. The face images of low faces are then stitched back to a background face video of the subject, resulting in video-realistic performance. Experiments on the LIPS 2008/2009 standard audio-visual corpus [35] show that the proposed LSTM-RNN approach significantly outperforms the state-of-the-art HMM-based approach.

2 Related work

Various talking head approaches have been proposed during the last decades. Broadly speaking, a talking head can be driven by text or speech, which depends largely on the application at hand. A text-driven talking head usually consists of a text-to-speech (TTS) module and a text-to-face-animation module. The facial animation is finally synchronized with the synthetic speech. Speech-driven talking head seems more straightforward: an input acoustic feature sequence is directly mapped to an output visual feature sequence that is used to drive a face model. AT&T Bell labs developed a text-driven talking head for interactive services [7]. Microsoft released a text-driven avatar in Engkoo online dictionary for language learning [41]. We proposed a real-time speech-driven talking avatar based on real-time phoneme recognition and motion trajectory generation through dynamic visemes [24].

According to the underlying face model, talking heads can be classified into model-based [1, 2, 19, 20, 26, 34, 43] and video-based [3, 7, 9, 25, 39, 46]. The model-based approaches usually animate the face using a deformable model while the video-based approaches concatenate short facial video clips from a prerecorded video dataset. Both approaches have merits and demerits. Model-based approaches are flexible because the face can be deformed in any reasonable ways, but they generally lack video-realism due to the complexity of real surfaces, textures and motions. In contrast, video-based approaches usually can achieve (near-) video realism performance because of the static nature of the texture, but they often lack flexibility as the facial animations are only confined to the limited samples in the video database.

Both text- and speech-driven taking heads desire an input to visual feature conversion or mapping algorithm. The conversion is not trivial because of the coarticulation, which causes a given phoneme to be pronounced differently depending on the surrounding phonemes. This phenomenon leads to an essential problem in talking head animation, namely *lip synchronization* or *lip-sync* for short. Lip-sync can be regarded as a regression or a classification task. Regression approaches try to directly map input features into continuous visual parameters, e.g., using linear regression and time-delayed neural network [28]. Classification approaches usually consider a phonetic representation, rather from acoustic speech recognition or text analysis module of a TTS engine, and generate visual parameters using mapping rules or concatenation of model parameters. The regression methods can generate continuous trajectories, but the facial motion is not accurate enough to provide useful visual information [18, 28]. Early classification approaches define a direct mapping from phonemes to basic mouth units, known as visemes, and mouth animation is achieved by simply morphing key images of these visemes [8]. The performance achieved is far from natural because coarticulation is not fully considered. Current main-stream classification methods usually learn an audio-to-visual mapping from a database in a statistical way, and thus the animation becomes more natural.

Hidden Markov models (HMMs) have shown tremendous success in modeling speech. Thus they have been recently investigated in talking head animation or visual speech synthesis [33, 45]. In an HMM-based talking head, audio and visual speech are jointly modeled by HMMs and audio/visual speech parameter trajectories are synthesized using the trajectory HMM algorithm under the maximum likelihood criterion [23]. One obvious drawback of the HMM-based talking head is its blurring animation due to the maximum likelihood-based statistical modeling. So there are some hybrid approaches that use the HMM predicted trajectory to guide the sample selection process [37, 39], which combines the advantages of both the video-based concatenation and the HMM-based modeling approaches. For both HMM-based parametric and HMM-guided hybrid approaches, the statistically trained

HMMs are crucial because the HMM-predicted visual trajectories determine how well the visual appearance can be rendered. Although HMM can model sequential data efficiently, there are still some well-known limitations [48], such as the wrong model assumptions out of necessity, e.g., Gaussian mixture model (GMM) and its diagonal covariance, and the greedy, hence suboptimal, search derived decision-tree based contextual state clustering.

Motivated by the deep neural network (DNN)'s superior performance in automatic speech recognition and other tasks, neural network approaches have been recently explored in a highly related task, i.e., speech synthesis [21, 31, 49]. There are several advantages of the deep NN-based synthesis approaches: it can model long-span, high dimensional and the correlation of input features; it is able to learn non-linear mapping between input and output with a deep-layered, hierarchical, feed-forward and recurrent structure; it has the discriminative and predictive capability in generation sense, with appropriate cost function(s), e.g. generation error. Recently, recurrent neural networks (RNNs) [42] and their bidirectional variant, bidirectional RNNs (BRNNs) [27], become popular because they are able to incorporate contextual information that is essential for sequential data modeling. Conventional RNNs cannot well model the long-span relations in sequential data because of the vanishing gradient problem [16]. Hochreiter et al. [17] found that the LSTM architecture, which uses purpose-built memory cells to store information, is better at exploiting long range context. Combining BRNNs with LSTM gives BLSTM, which can access long-range context in both directions.

Hence in this paper, we propose to use LSTM-RNNs as a regression model to learn a direct mapping from input audio/text to the output visual parameters. This avoids the problematic decision tree clustering in HMM-based approaches. The decision tree subdivides the model space by a hard division based on one feature at a time, fragmenting the data and failing to exploit interactions between linguistic context features [48]. Similar to the HMM trajectory guided sample selection approach [37], to achieve video-realism, the visual parameter trajectory is further used as guidance in a video sample selection procedure. The proposed LSTM-RNN approach can first give a convincing estimation of the visual trajectory given any unseen acoustic input, and then *tiles* the predicted trajectories with real image samples. Therefore the animated talking head achieves both lip synchronization and video-realistic and experiments have shown its superior performance as compared with the HMM-based approach.

3 System overview

Figure 1 shows the diagram of the proposed photo-real talking head that is composed of a training stage and a synthesis stage. First of all, this framework requires an audio/visual database of a subject talking to a camera with frontal view as the training data. In this study, we employ the LIPS 2008/2009 Visual Speech Synthesis Challenge data [35] to build our talking head system. Please note that this is a language independent framework and it can be easily extended to another subject speaking a different language.

In the training stage, the input text and audio are converted into a sequence of input features **I**. As mentioned above, the input features can be contextual labels, acoustic features or the combination of them, each corresponding to text-driven, speech-driven and text/speech-driven talking head, respectively. The lower face image sequence is transformed into a sequence of output features **O** and this step is realized by active appearance model (AAM) of face. A small set of video data is sufficient to build a high-quality talking head with our proposed method. In our approach, the footage of the video for training is about 17 minutes.

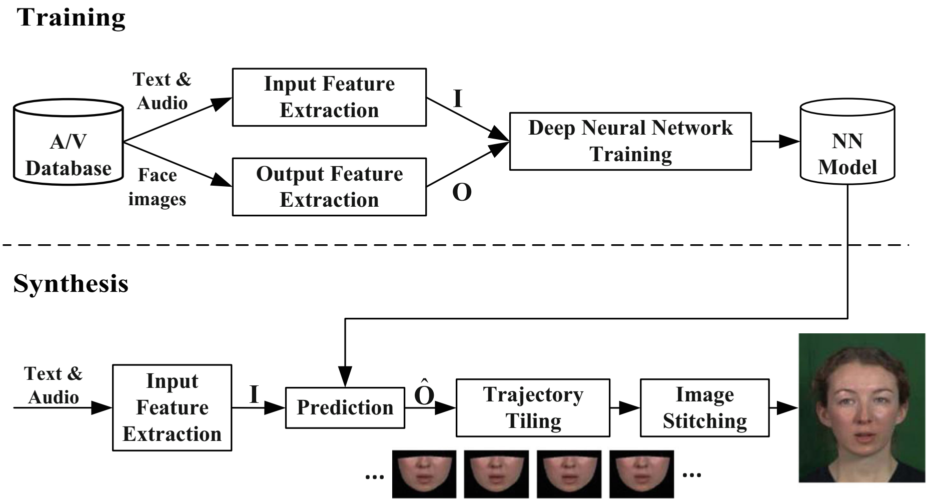


Fig. 1 System overview of the proposed talking head

Given the input **I** and the output **O**, we train a deep neural network, e.g., LSTM-RNN, to learn the regression model by minimizing the prediction error of **O**.

In the synthesis stage, for any input speech and text, we first extract the input sequence **I** and then predict the visual AAM parameters $\hat{\mathbf{O}}$ using the well-trained deep neural network. After that, we use the predicted visual parameter sequence $\hat{\mathbf{O}}$ as a guidance to select the most-likely image sequence from the database. Finally, we stitch the selected image sequence to a pre-defined background whole-face image sequence, resulting in lip-synced video-realistic talking head animation.

4 Feature representation

4.1 Input feature I

The input of a desired talking head system can be any arbitrary text along with natural audio recordings or TTS-synthesized speech. For speech recordings, the phoneme/state time alignment can be obtained by conducting forced alignment using a trained phoneme recognizer. For TTS synthesized speech, the phoneme/state sequence and time offset are just a by-product of the synthesis process. Specifically, the input feature can be contextual labels **L**, acoustic feature **A** or the combination of them.

For contextual labels **L**, we convert the phoneme/state sequence and their time offset into a label sequence, denoting as $\mathbf{L} = (\mathbf{l}_1, \ldots, \mathbf{l}_t, \ldots, \mathbf{l}_T)$, where T is the number of frames in the sequence. The format of the frame-level label $\mathbf{l}_t$ uses the one-hot representation, i.e., one vector for each frame, shown as follows:

$$[\underbrace{0, \ldots, 0, \ldots, 1}_{K}, \underbrace{1, \ldots, 0, \ldots, 0}_{K}, \underbrace{0, 0, 1, \ldots, 0}_{K}, \underbrace{0, 1, 0}_{3},$$

where K denotes the number of phonemes in the given language. We use triphone plus the information of three states to identify $\mathbf{l}_t$. The first 3 K-element sub-vectors denote the identities of the left, current and right phonemes in the triphone, respectively, and the last 3 elements represent the phoneme state which can be obtained from forced alignment of a natural speech recording or a TTS system. The reason why state information is included in the labels is to improve the distinguishability. If not included, the contextual label of the current phoneme will keep constant in its duration which will lead to lack of expressiveness of the lip movements.

Please note that the contextual label can be easily extended to contain richer information, like positions in syllables, words, stress, part-of-speech, etc. Due to the limitation of the training data, in our experiment we only consider phoneme and state level labels.

For acoustic feature **A**, we use standard Mel-frequency cepstral coefficients (MFCCs) and their delta and delta-delta coefficients. The dimensionality of **A** is 39 in our work.

4.2 Output feature O

Most of the speech-originated facial movements are constrained to the lower part of a human face. Hence the output of our talking head is a visual stream which is a sequence of lower face images strongly correlated to the underlying speech. As a raw face image is hard to model directly due to the high dimensionality, we use active appearance model (AAM) for visual feature extraction. AAM is a joint statistical model compactly representing both the shape and the texture variations and the correlation between them [5].

Since the head movements, raised by natural speaking, may hinder the face modeling, we perform head pose normalization among all the face images before AAM modeling. With the help of an effective 3D model-based head pose tracking algorithm [38], the head pose in each image frame is normalized to a fully frontal view and further aligned.

The shape of the j-th lower face, $\mathbf{s}_j$, can be represented by the concatenation of the x and y coordinates of N facial feature points:

$$\mathbf{s}_j = (x_{j1}, x_{j2}, \ldots, x_{jN}, y_{j1}, y_{j2}, \ldots, y_{jN}), \tag{1}$$

where $j = 1, 2, \ldots, J$ and J is the total number of the face images. In this work, we use a set of 48 typical facial feature points, as shown in Figure 2a. The mean shape is simply defined by

$$\mathbf{s}_0 = \sum\nolimits_{j=1}^{J} \mathbf{s}_j / J. \tag{2}$$

Applying principal component analysis (PCA) to all J shapes, $\mathbf{s}_j$ can be given approximately by:

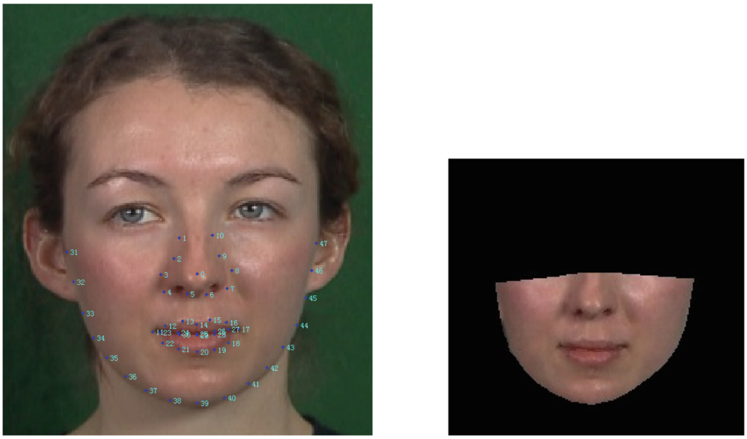
$$\mathbf{s}_j = \mathbf{s}_0 + \sum\nolimits_{i=1}^{N_{\text{shape}}} a_{ji} \tilde{\mathbf{s}}_i = \mathbf{s}_0 + \mathbf{a}_j \mathbf{P}_s, \tag{3}$$

where $\mathbf{P}_s = [\tilde{\mathbf{s}}_1, \tilde{\mathbf{s}}_2, \ldots, \tilde{\mathbf{s}}_i, \ldots, \tilde{\mathbf{s}}_{N_{\text{shape}}}^{\top}$ denotes the eigenvectors corresponding to the N_{shape} largest eigenvalues and $\mathbf{a}_j = [a_{j1}, a_{j2}, \ldots, a_{ji}, \ldots, a_{jN_{\text{shape}}}$ is the j-th shape parameter vector.

We define the inner triangulation structure by applying Delaunay Triangulation on the mean shape $\mathbf{s}_0$. For an arbitrary shape, we can establish a unique transformation between this shape and the mean shape. Then the texture of the j-th face image, $\mathbf{t}_j$, can be defined by a vector concatenating the RGB value of every pixel that lies inside the mean shape $\mathbf{s}_0$:

$$\mathbf{t}_j = (r_{j1}, r_{j2}, \ldots, r_{jU}, g_{j1}, g_{j2}, \ldots, g_{jU}, b_{j1}, b_{j2}, \ldots, b_{jU}), \tag{4}$$

where $j = 1, 2, \ldots, J$ and U is the total number of pixels in the lower face region.



(a) The 48 facial feature points.

(b) The texture of a lower face.

Fig. 2 Facial feature points and the texture of a *lower* face

As the dimensionality of the texture vector is too high to use PCA directly, we apply EMPCA to all J textures [32]. As a result, the j-th texture $\mathbf{t}_j$ can be given approximately by:

$$\mathbf{t}_j = \mathbf{t}_0 + \sum\nolimits_{i=1}^{N_{\text{texture}}} b_{ji} \tilde{\mathbf{t}}_i = \mathbf{t}_0 + \mathbf{b}_j \mathbf{P}_t, \tag{5}$$

where $\mathbf{t}_0$ is the mean texture, $\mathbf{P}_t$ contains the orthonormal eigenvectors corresponding to the N_{texture} largest eigenvalues, and $\mathbf{b}_j$ is the j-th texture parameter vector.

We simplify each elements' distributions of all the shape and texture parameters into normal distributions. The corresponding means and standard deviations are $a_0^{i_1}$, $b_0^{i_2}$ and $\sigma_{a^{i_1}}$, $\sigma_{b^{i_2}}$ (i_1 and i_2 are the indices of arbitrary elements). If an arbitrary shape parameter $a_k^{i_1}$ and a texture parameter $b_k^{i_2}$ are in conformity with the training data, they should meet the following conditions:

$$\left\| a_k^{i_1} - a_0^{i_1} \right\| \leq 3\sigma_{a^{i_1}}, i_1 = 1, 2, \ldots, N_{\text{shape}}, \tag{6}$$

$$\left\| b_k^{i_2} - b_0^{i_2} \right\| \leq 3\sigma_{b^{i_2}}, i_2 = 1, 2, \ldots, N_{\text{texture}}. \tag{7}$$

The above shape and texture models can only control the shape and texture separately. In order to recover the correlation between the shape and the texture, $\mathbf{a}_j$ and $\mathbf{b}_j$ are combined in another round of PCA as follows:

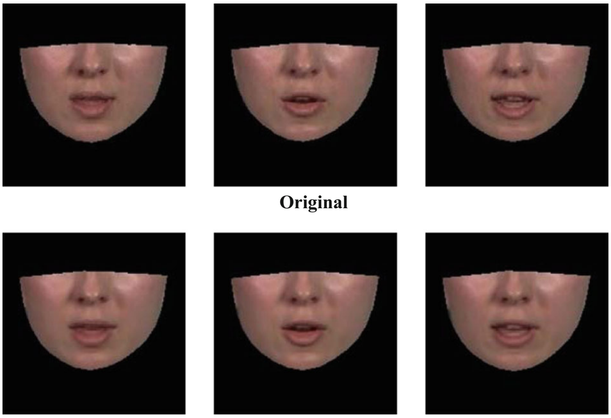
$$(\mathbf{a}_j, \mathbf{b}_j) = \sum\nolimits_{i=1}^{N_{\text{appearance}}} o_{ji} \tilde{\mathbf{o}}_i = \mathbf{o}_j \mathbf{P}_o, \tag{8}$$

assuming that $\mathbf{P}_{os}$ and $\mathbf{P}_{ot}$ are formed by extracting the first N_{shape} and the last N_{texture} values from each component in $\mathbf{P}_o$. Simply combining the above equations gives rise to:

$$\mathbf{s}_j = \mathbf{s}_0 + \mathbf{o}_j \mathbf{P}_{os} \mathbf{P}_s = \mathbf{s}_0 + \mathbf{o}_j \mathbf{Q}_s, \tag{9}$$

$$\mathbf{t}_j = \mathbf{t}_0 + \mathbf{o}_j \mathbf{P}_{ot} \mathbf{P}_t = \mathbf{t}_0 + \mathbf{o}_j \mathbf{Q}_t. \tag{10}$$

Now, we can reconstruct the shape and texture of the j-th lower face image by only one appearance parameter vector $\mathbf{o}_j$. Figure 3 shows some original lower-face images and



Original

Reconstructed

Fig. 3 Some original lower-face images and their reconstructed ones from the AAM

their reconstructed ones from the AAM. We can see that the reconstructed images are quite similar to the original ones. Subsequently, the lower face sequence with T frames can be represented by the output feature sequence $\mathbf{O} = (\mathbf{o}_1, \ldots, \mathbf{o}_t, \ldots, \mathbf{o}_T)$.

5 LSTM-RNN based talking head

Motivated by its superior performance in many tasks, we propose to use LSTM-RNN for audio-to-visual mapping. In this section, we first briefly review the basics about RNN and RNN with LSTM cells. After that, the LSTM-RNN based talking head will be introduced.

5.1 RNN

Allowing cyclical connections in a feed-forward neural network, we obtain recurrent neural networks (RNNs) [42]. Different from the feed-forward ones, RNNs are able to incorporate contextual information from previous input vectors, which allows them to remember past inputs and persist in the network's internal state. This property makes them an attractive choice for sequence-to-sequence learning. For a given input vector sequence $\mathbf{x} = (x_1, x_2 \ldots, x_T)$, the forward pass of RNNs is as follows:

$$\mathbf{h}_t = \mathcal{H}(\mathbf{W}_{xh}x_t + \mathbf{W}_{hh}h_{t-1} + b_h), \tag{11}$$

$$y_t = \mathbf{W}_{hy}h_t + b_y, \tag{12}$$

where $t = 1, \ldots, T$, and T is the length of the sequence; $\mathbf{h} = (h_1, \ldots, h_T)$ is the hidden state vector sequence computed from $\mathbf{x}$; $\mathbf{y} = (y_1, \ldots, y_T)$ is the output vector sequence; $\mathbf{W}$ is the weight matrices, where $\mathbf{W}_{xh}$, $\mathbf{W}_{hh}$ and $\mathbf{W}_{hy}$ are the input-hidden, hidden-hidden and hidden-output weight matrices, respectively. b_h and b_y are the hidden and output bias vectors, respectively and $\mathcal{H}$ denotes the nonlinear activation function for hidden nodes.

For our talking head system, because of the speech coarticulation phenomenon, we desire the model to have access to both past and future context. But conventional RNNs can only access the past context and they ignore the future context. So the bidirectional recurrent neural networks (BRNNs), as shown in Fig. 4, are used to relieve this problem. BRNNs

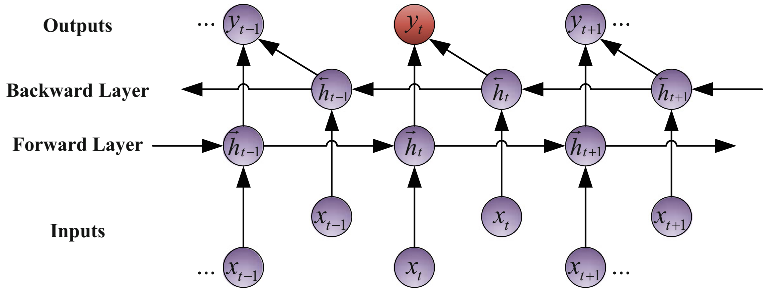


Fig. 4 Bidirectional recurrent neural networks (BRNNs)

compute both forward state sequence $\overrightarrow{\mathbf{h}}$ and backward state sequence $\overleftarrow{\mathbf{h}}$, as formulated below:

$$\overrightarrow{\mathbf{h}}_t = \mathcal{H}(\mathbf{W}_{x\overrightarrow{h}} x_t + \mathbf{W}_{\overrightarrow{h}\overrightarrow{h}} \overrightarrow{h}_{t-1} + b_{\overrightarrow{h}}), \tag{13}$$

$$\overleftarrow{\mathbf{h}}_t = \mathcal{H}(\mathbf{W}_{x\overleftarrow{h}} x_t + \mathbf{W}_{\overleftarrow{h}\overleftarrow{h}} \overleftarrow{h}_{t-1} + b_{\overleftarrow{h}}), \tag{14}$$

$$y_t = \mathbf{W}_{\overrightarrow{h}y} \overrightarrow{h}_t + \mathbf{W}_{\overleftarrow{h}y} \overleftarrow{h}_t + b_y. \tag{15}$$

5.2 LSTM-RNN

Conventional RNNs can access only a limited range of context because of the vanishing gradient problem. Long short-term memory (LSTM) [17] uses purpose-built memory cells, as shown in Fig. 5, to store information which is designed to overcome this limitation. In sequence-to-sequence mapping tasks, LSTM has been shown capable of bridging very long time lags between input and output sequences by enforcing constant error flow. For LSTM, the recurrent hidden layer function $\mathcal{H}$ is implemented as follows:

$$i_t = \sigma(\mathbf{W}_{xi} x_t + \mathbf{W}_{hi} h_{t-1} + \mathbf{W}_{ci} c_{t-1} + b_i), \tag{16}$$

$$f_t = \sigma(\mathbf{W}_{xf} x_t + \mathbf{W}_{hf} h_{t-1} + \mathbf{W}_{ci} c_{t-1} + b_f), \tag{17}$$

$$a_t = \tau(\mathbf{W}_{xc} x_t + \mathbf{W}_{hc} h_{t-1} + b_c), \tag{18}$$

$$c_t = f_t c_{t_1} + i_t a_t, \tag{19}$$

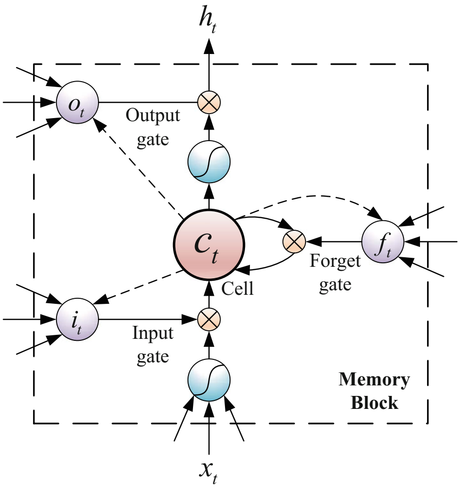
$$o_t = \sigma(\mathbf{W}_{xo} x_t + \mathbf{W}_{ho} h_{t-1} + \mathbf{W}_{co} c_t + b_o), \tag{20}$$

$$h_t = o_t \theta(c_t), \tag{21}$$

where σ is the sigmoid function; i, f, o, a and c are input gate, forget gate, output gate, cell input activation and cell memory, respectively. τ and θ are the cell input and output non-linear activation functions, in which tanh is generally chosen. The multiplicative gates allow LSTM memory cells to store and access information over long periods of time, thereby avoiding the vanishing gradient problem.

Combining BRNNs with LSTM gives rise to BLSTM, which can access long-range context in both directions. Motivated by the success of deep neural network architectures, we propose to use deep BLSTM-RNNs (DBLSTM-RNNs) to establish the audio-to-visual mapping in our talking head system. Deep BLSTM-RNN is created by stacking multiple BLSTM hidden layers.

Fig. 5 Long short-term memory (LSTM)



5.3 Deep BLSTM-RNNs based talking head

The extracted input sequence **I** and output feature sequence **O** are two time varying parallel sequences. After resampling, we can easily make the two sequences at the same frame rate. In particular, at the t-th frame, the input of the network is the t-th input vector $\mathbf{i}_t$ and the output is the t-th output feature vector $\mathbf{o}_t$. As described in [11], the basic idea of this bidirectional structure is to present each sequence forwards and backwards to two separate recurrent hidden layers, both of which are connected to the same output layer. This provides the network with complete, symmetrical, past and future context for every point in the input sequence. Please note that in Fig. 6, more hidden layers can be added in to construct a deep BLSTM-RNN.

In the training stage, we have multiple sequence pairs of **I** and **O**. As we represent both sequences as continuous numerical vectors, the network is treated as a regression model minimizing the sum of squared errors (SSE) of predicting $\hat{\mathbf{O}}$ from **I**. In the test (or synthesis) stage, given any arbitrary text along with natural or synthesized speech, we firstly convert them into a sequence of input features, then feed into the trained network, and the output of the network is the predicted visual AAM feature sequence. Please note that the maximum likelihood parameter generation (MLPG) algorithm has not been used in the parameter generation process because the DBLSTM approach can generate smooth trajectories without the smoothing step. This has also been shown in a previous TTS work [10]. After reconstructing the AAM feature vectors to RGB images, we can get the video realistic image sequence of the lower face. Finally, we stitch the lower face to a background face and render the facial animation of the talking head.

Learning deep BLSTM network can be regarded as optimizing a differentiable error function:

$$E(\mathbf{w}) = \sum\nolimits_{k=1}^{M_{\text{train}}} E_k(\mathbf{w}), \tag{22}$$

where M_{train} represents the number of sequences in the training data and **w** denotes the network inter-node weights. In our task, the training criterion is to minimize the SSE

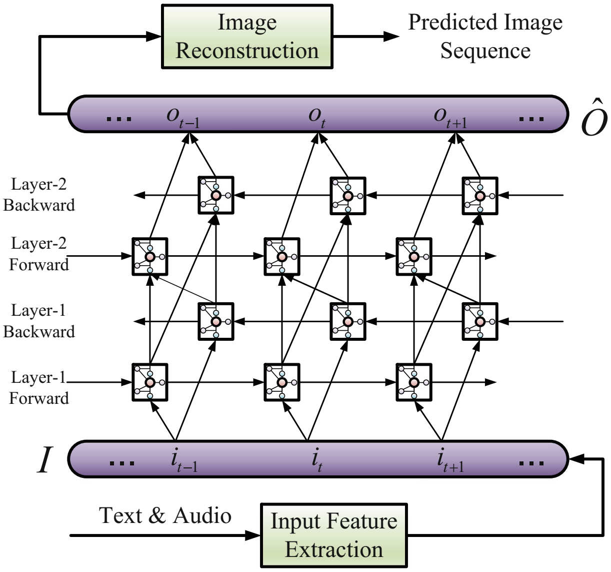


Fig. 6 DBLSTM-RNNs in our talking head system

between the predicted visual features $\hat{\mathbf{O}} = (\hat{\mathbf{o}}_1, \hat{\mathbf{o}}_2, \ldots, \hat{\mathbf{o}}_T)$ and the ground truth $\mathbf{O} = (\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_T)$. For a particular input sequence k, the error function takes the form:

$$E_k(\mathbf{w}) = \sum\nolimits_{t=1}^{T_k} E_{kt} = \frac{1}{2}\sum\nolimits_{t=1}^{T_k} \left\| \hat{\mathbf{o}}_t^k - \mathbf{o}_t^k \right\|^2, \tag{23}$$

where T_k is the total number of frames in the k-th sequence. In every iteration, we calculate the error gradient with the following equation:

$$\Delta\mathbf{w}(r) = m\Delta\mathbf{w}(r-1) - \alpha\frac{\partial E(\mathbf{w}(r))}{\partial\mathbf{w}(r)}, \tag{24}$$

where $0 \leq \alpha \leq 1$ is the learning rate, $0 \leq m \leq 1$ is the momentum parameter, and $\mathbf{w}(r)$ represents the vector of weights after r-th iteration of update. The convergence condition is that the validation error has no obvious change after R iterations.

We use back-propagation through time (BPTT) algorithm to train the network. In the BLSTM hidden layer, BPTT is applied to both forward and backward hidden nodes and back-propagates layer by layer. Take the error function derivatives with respect to the output of the network as an example. For $\hat{\mathbf{o}}_t^k = (\hat{o}_{t1}^k, \ldots, \hat{o}_{tj}^k, \ldots, \hat{o}_{tN_{\text{appearance}}}^k)$ in the k-th $\hat{\mathbf{O}}$, because the activation function used in the output layer is an identity function, we have

$$\hat{o}_{tj}^k = \sum_h w_{oh} z_{ht}^k, \tag{25}$$

where o is the index of the an output node, z_{ht}^k is the activation of a node in the hidden layer connected to the node o, and w_{oh} is the weight associated with this connection. By applying the chain rule for partial derivatives, we can obtain

$$\frac{\partial E_{kt}}{\partial w_{oh}} = \sum\nolimits_{j=1}^{N_{\text{appearance}}} \frac{\partial E_{kt}}{\partial \hat{o}_{tj}^k} \frac{\partial \hat{o}_{tj}^k}{\partial w_{oh}}, \tag{26}$$

and according to (23) and (25), we can derive

$$\frac{\partial E_{kt}}{\partial w_{oh}} = \sum\nolimits_{j=1}^{N_{\text{appearance}}} (\hat{o}_{tj}^{k} - o_{tj}^{k}) z_{ht}^{k}, \tag{27}$$

$$\frac{\partial E_{k}}{\partial w_{oh}} = \sum\nolimits_{t=1}^{T} \frac{\partial E_{kt}}{\partial w_{oh}}. \tag{28}$$

6 Trajectory tiling for video-realistic talking head

Given any arbitrary text along with natural or synthesized speech, we can generate the corresponding lower face image sequence reconstructed from the AAM using the above mentioned DBLSTM-RNN talking head system. Although the predicted image sequence is smooth and robust to the coarticulation phenomenon, it still suffers from the *blur* problem that may be caused by dimensionality reduction in AAM, statistical modeling and trajectory generation. To solve the blur problem, inspired by a previous work [39], we use a trajectory tiling approach to achieve a video-realistic talking head. Specifically, we use the predicted AAM visual parameter sequence $\hat{\mathbf{O}}$ as a guidance to select the most likely image sequence from the database. After that, we stitch the selected image sequence to a pre-defined background sequence, and finally we can generate a lip-synced video-realistic talking head.

6.1 Cost function

Motivated by the unit selection in concatenative speech synthesis, the cost function is defined as the weighted sum of the target and concatenation costs:

$$C(\hat{\mathbf{O}}, \hat{\mathbf{R}}) = \sum\nolimits_{i=1}^{T} w^{t} C^{t}(\hat{\mathbf{o}}_i, \hat{\mathbf{r}}_i) + \sum\nolimits_{i=2}^{T} w^{c} C^{c}(\hat{\mathbf{r}}_{i-1}, \hat{\mathbf{r}}_i), \tag{29}$$

where $\hat{\mathbf{O}} = (\hat{\mathbf{o}}_1, \ldots, \hat{\mathbf{o}}_i, \ldots, \hat{\mathbf{o}}_T)$ and $\hat{\mathbf{R}} = (\hat{\mathbf{r}}_1, \ldots, \hat{\mathbf{r}}_i, \ldots, \hat{\mathbf{r}}_T)$ are predicted output sequence and selected real image sequence, respectively, and w^t and w^c are target weight and concatenation weight, respectively.

The selected image sequence should be similar to the trajectory as much as possible because the procedure of selection is guided by the trajectory. Hence the target cost between an image sample $\hat{\mathbf{r}}_i$ and predicted visual feature $\hat{\mathbf{o}}_i$ is defined by the Euclidean distance:

$$C^{t}(\hat{\mathbf{o}}_i, \hat{\mathbf{r}}_i) = \left\| \hat{\mathbf{o}}_i - \mathbf{o}_i \right\|, \tag{30}$$

where $\mathbf{o}_i$ is the ground truth AAM visual feature vector corresponding to $\hat{\mathbf{r}}_i$.

As the coarticulation phenomenon, we must take the smoothness of the concatenation between adjoining selected images into consideration. We use cosine distance to measure the concatenation cost between two image samples $\hat{\mathbf{r}}_i$ and $\hat{\mathbf{r}}_j$:

$$cos(\hat{\mathbf{r}}_i, \hat{\mathbf{r}}_j) = \frac{\mathbf{o}_i \cdot \mathbf{o}_j}{\|\mathbf{o}_i\| \, \|\mathbf{o}_j\|}. \tag{31}$$

Assuming that the corresponding samples of $\hat{\mathbf{r}}_i$ and $\hat{\mathbf{r}}_j$ in the sample library are $\mathbf{r}_p$ and $\mathbf{r}_q$, i.e., $\hat{\mathbf{r}}_i = \mathbf{r}_p$, and $\hat{\mathbf{r}}_j = \mathbf{r}_q$, where, p and q are the sample indices in video recording. Hence, $\mathbf{r}_p$ and $\mathbf{r}_{p+1}$, $\mathbf{r}_q$ and $\mathbf{r}_{q-1}$ are consecutive frames in the original recording. The next

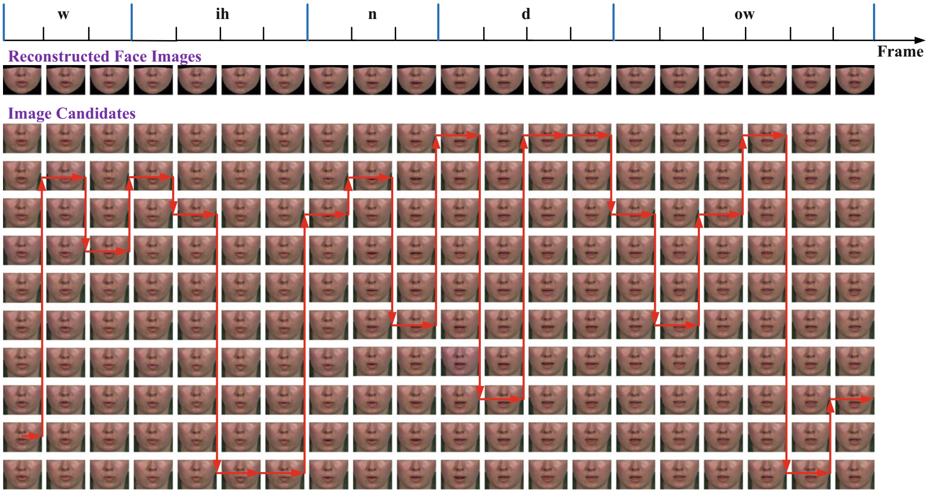


Fig. 7 Illustration for trajectory tiling in video-realistic talking head. In this example, the word 'window' is spoken by the speaker. The reconstructed face images are generated by the predicted AAM visual features. They serve as the guidance to select the optimal image sequence from the image candidates (K=10 at each time)

image of $\mathbf{r}_p$ is $\mathbf{r}_{p+1}$ and the former image of $\mathbf{r}_q$ is $\mathbf{r}_{q-1}$. We define the concatenation cost between $\hat{\mathbf{r}}_i$ and $\hat{\mathbf{r}}_j$ as:

$$C^c(\hat{\mathbf{r}}_i, \hat{\mathbf{r}}_j) = C^c(\mathbf{r}_p, \mathbf{r}_q) = 1 - \frac{1}{2}\left[\cos(\mathbf{r}_p, \mathbf{r}_{q-1}) + \cos(\mathbf{r}_{p+1}, \mathbf{r}_q)\right]. \tag{32}$$

Because $\cos(\mathbf{r}_p, \mathbf{r}_p) = \cos(\mathbf{r}_q, \mathbf{r}_q) = 1$,

$$C^c(\mathbf{r}_p, \mathbf{r}_{p+1}) = C^c(\mathbf{r}_{q-1}, \mathbf{r}_q) = 0, \tag{33}$$

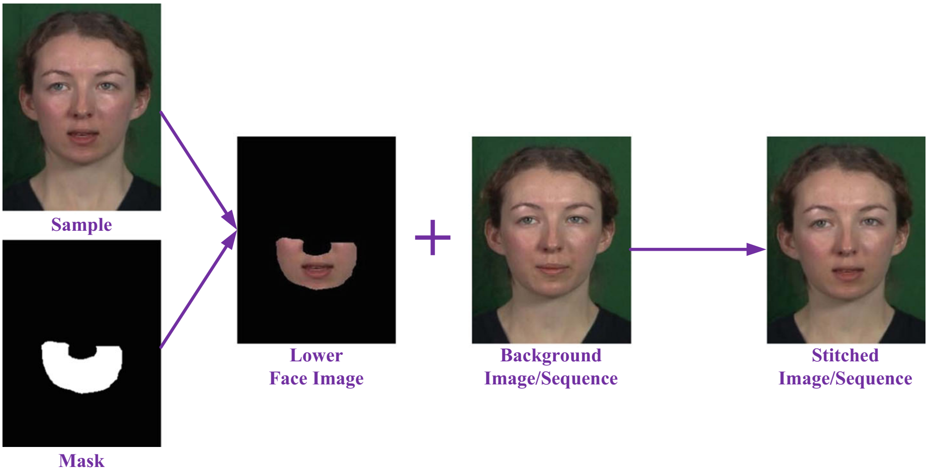


Fig. 8 Illustration of the image stitching process

Table 1 Network topologies tested in our experiments

Hidden layer	F, B, BB, BF, FB, FF, BBB, BBF, BFB BFF, FBB, FBF, FFB and FFF
Node	64, 128, 256 and 512

which means that the concatenation cost encourages the selection of consecutive frames in the original recording. In other words, this selection algorithm prefers to select real video segments because of zero concatenation cost.

We select the image sequence that makes the cost function reach the minimum value. Because there are thousands of images in the training data, we use Viterbi search to find the optimal image sequence. This is done by two pruning steps. Firstly, for every target frame in the trajectory, K-nearest samples are selected as candidates according to the target cost. K is set to 40 in our study. Secondly, the best image sequence is chosen by Viterbi search in terms of concatenation cost. This pruning procedure is depicted in Fig. 7 and K is set to 10 here to simply illustrate this procedure.

6.2 Image stitching

We use the same strategy in [6] to stitch the lower face images back onto the full face images. Local deformations are required in order to stitch the shape of the mouth and jaw line correctly to avoid the artifacts possibly caused when the stitches are across the jaw line. After local deformation around the jaw line, we use Poisson image editing technique [29]

Table 2 The objective experimental results for networks with different hidden layers and numbers of nodes for the text-driven talking head

Node	64		128		256		512	
TP	RMSE	CORR	RMSE	CORR	RMSE	CORR	RMSE	CORR
B	70.499	0.543	70.227	0.546	71.177	0.535	71.309	0.532
F	76.087	0.418	75.864	0.421	76.666	0.403	75.954	0.418
BB	68.952	0.569	69.297	0.563	69.873	0.552	71.121	0.535
BF	69.222	0.567	69.580	0.563	70.610	0.541	70.706	0.543
FB	67.899	0.584	67.926	0.583	67.641	0.586	68.194	0.580
FF	75.841	0.424	76.073	0.417	75.679	0.426	75.453	0.430
BBB	68.232	0.578	68.506	0.574	68.882	0.569	70.596	0.544
BBF	68.135	0.580	69.012	0.567	68.882	0.569	69.947	0.552
BFB	68.075	0.582	67.672	0.586	68.580	0.573	69.441	0.560
BFF	69.174	0.564	69.478	0.560	70.896	0.538	70.598	0.542
FBB	**67.144**	**0.593**	**66.971**	**0.597**	67.133	0.593	67.760	0.584
FBF	67.433	0.589	67.241	0.591	**67.117**	**0.594**	67.947	0.582
FFB	67.503	0.589	67.668	0.585	67.762	0.583	**67.464**	**0.589**
FFF	75.788	0.424	75.847	0.422	75.655	0.427	76.249	0.413

to blend the lower face images onto the full face images automatically. The editing region is specified by a mouth replacement mask, as shown in Fig. 8. We randomly selected a consecutive video segment as the background. Because the selected lower face images are smooth and realistic in terms of the pre-defined cost function, we can finally generate a natural and smooth talking head, as shown in Fig. 8.

7 Experiments

7.1 Experimental setup

Our experiments were carried out on the standard LIPS 2008/2009 Visual Speech Synthesis Challenge data [35]. The database contains 278 audio-video files with English sentences spoken by a native female speaker in a neutral style. The full contextual labels are generated with a phoneme dictionary which has 50 phonemes. The frame rate of the video files is 50 fps and all together 61,028 face images with pixel resolution 288 × 360 are collected. We randomly divided the database into 3 disjoint parts: 80 % for training, 10 % for validation and 10 % for testing. We randomly selected 20,000 images from the training set for lower face AAM modeling. We chose top 76 shape and 100 texture principal components containing about 85.7 % cumulative energy contents, respectively. The final dimension of the output appearance vector ($\hat{\mathbf{o}}_t$) is 91. We find that the use of more principal components will not lead to further performance improvement. In the neural network training, we set the learning rate and the momentum to 1e-6 and 0.9, respectively and the weights were initialized with a Gaussian random distribution.

Table 3 The objective experimental results for networks with different hidden layers and numbers of nodes for the speech-driven talking head

Node	64		128		256		512	
TP	RMSE	CORR	RMSE	CORR	RMSE	CORR	RMSE	CORR
B	71.104	0.524	70.535	0.531	70.094	0.540	69.402	0.556
F	82.319	0.257	80.969	0.303	81.701	0.279	82.412	0.256
BB	69.477	0.557	68.603	0.572	**68.534**	**0.574**	69.356	0.554
BF	69.602	0.553	70.653	0.535	70.046	0.545	69.288	0.562
FB	71.227	0.524	71.622	0.516	69.997	0.548	71.213	0.522
FF	78.821	0.360	79.201	0.351	79.259	0.348	78.422	0.373
BBB	68.324	0.576	68.573	0.568	68.587	0.569	69.494	0.556
BBF	**68.026**	**0.577**	68.959	0.569	68.572	0.570	**68.484**	**0.572**
BFB	69.458	0.556	**68.192**	**0.577**	68.574	0.572	69.694	0.559
BFF	70.945	0.528	69.082	0.557	69.448	0.554	69.784	0.552
FBB	69.337	0.560	69.684	0.553	69.144	0.555	69.205	0.556
FBF	70.680	0.531	69.902	0.545	70.967	0.525	69.432	0.558
FFB	71.939	0.512	72.154	0.507	70.732	0.524	68.690	0.571
FFF	80.047	0.320	79.726	0.330	79.288	0.345	79.384	0.346

7.2 Objective evaluation

We conducted objective evaluations by directly comparing the predicted visual AAM features with the ground truth AAM parameters. Two objective metrics are used, defined as follows:

$$RMSE = \frac{\sum_{k=1}^{M_{\text{test}}} \sum_{t=1}^{T_k} \sqrt{\parallel \hat{\mathbf{o}}_t^k - \mathbf{o}_t^k \parallel^2 / N_{\text{appearance}}}}{\sum_{k=1}^{M_{\text{test}}} T_k}, \tag{34}$$

$$CORR = \frac{\sum_{k=1}^{M_{\text{test}}} \sum_{t=1}^{T_k} corr(\hat{\mathbf{o}}_t^k, \mathbf{o}_t^k)}{\sum_{k=1}^{M_{\text{test}}} T_k}, \tag{35}$$

where M_{test} represents the number of sequences in the test data, T_k is the length of the kth sequence in the test data and $\mathbf{o}_t^k$ is the tth frame of the output feature in the kth output sequence. $corr(\hat{\mathbf{o}}_t^k, \mathbf{o}_t^k)$ denotes the correlation coefficient. Note that lower RMSE and higher CORR correspond to better performance.

The input features can be contextual labels, acoustic features or the combination of them. For the three kinds of feature inputs, we tested the performance of network topologies with different hidden layers (F–feed forward, B–BLSTM) and numbers of nodes, as described in Table 1. The results are summarized in Tables 2, 3 and 4, respectively. Please note that we have tested topologies with more than three layers. While the training data used in our study is quite limited, we found that the increase of the network layers does no help to the performance gain but leads to over-fitting. Moreover, the training time of the networks is increased dramatically.

Table 4 The objective experimental results for networks with different hidden layers and numbers of nodes for text-and-speech-driven talking head

Node	64		128		256		512	
TP	RMSE	CORR	RMSE	CORR	RMSE	CORR	RMSE	CORR
B	69.204	0.564	69.939	0.553	71.086	0.534	71.510	0.528
F	75.724	0.432	76.032	0.423	75.948	0.425	76.328	0.423
BB	69.388	0.560	69.360	0.564	69.385	0.562	71.448	0.528
BF	69.817	0.555	69.659	0.557	69.194	0.571	70.472	0.547
FB	66.904	0.599	66.474	0.602	68.116	0.581	67.724	0.588
FF	75.138	0.442	75.143	0.442	74.383	0.450	75.128	0.444
BBB	67.607	0.589	68.093	0.582	68.853	0.573	71.659	0.523
BBF	67.867	0.587	67.458	0.592	69.360	0.565	69.181	0.568
BFB	67.760	0.585	66.993	0.597	67.490	0.593	69.368	0.566
BFF	69.272	0.564	69.068	0.568	68.824	0.577	69.698	0.558
FBB	66.761	0.598	66.261	0.608	66.950	0.595	67.377	0.590
FBF	66.733	0.599	**66.242**	**0.608**	66.667	0.602	67.057	**0.598**
FFB	**66.722**	**0.601**	66.890	0.596	**66.454**	**0.602**	**66.907**	0.595
FFF	74.949	0.449	75.126	0.443	75.777	0.428	75.720	0.432

Table 5 Comparison between DBLSTM-RNN-based and HMM-based talking heads

Comparison	RMSE	CORR
HMM	71.414	0.496
DBLSTM	**67.144**	**0.593**

Results show that, in most cases, the 3-hidden-layer structures outperform the 1- and 2-hidden-layer structures and the structures with a BLSTM layer apparently outperform those without one. The best results are marked with bold font in these three tables. For text-driven talking head, in terms of the two objective metrics, FBB, FBF and FFB show superior performance. The best text-driven performance (RMSE=66.971, CORR=0.597) is achieved by FBB-128, a network with two BLSTM layers sitting on top of one feed-forward layer. For speech-driven talking head, BBF and BFB usually give good results and the best network topology is BBF with 64 nodes per layer. We notice that text-driven shows slightly better performance as compared with speech-driven. A possible explanation is that the same utterance spoken by the same person at various times may be more or less different, while the corresponding text is definitely the same, i.e., the input features in the text-driven system are more stable than that in the speech-driven system. When the input is the combination of text and speech (text-and-speech-driven), further performance gain is achieved and the best topology is FBF with 128 nodes per layer (RMSE=66.242, CORR=0.608). We believe that, although the improvement is limited, the complimentary information from the two modalities help to achieve this. Please note that, for a sanity check, we have tried unidirectional LSTM-RNNs. In general, the performance of unidirectional LSTM-RNN is not as good as the bidirectional one. Some recent studies, e.g., [10], have also shown that the bidirectional structure is essential for the success of sequential modeling of speech data.

7.3 DBLSTM-RNNs vs. HMM

We also compared our DBLSTM-RNN approach with the state-of-the-art HMM-based approach [37]. In the HMM-based system, five-state, left-to-right HMM triphone models were used, where each state was modeled by a single Gaussian with diagonal covariance. The HMMs were first trained in the maximum likelihood (ML) sense and then refined by the minimum generation error (MGE) training. We chose the best topology of text-driven talking head, i.e., FBB128, to compare with the HMM-based one. The results for the FBB128 deep BLSTM-RNN and the HMM are shown in Table 5. We can clearly see that the deep BLSTM-RNNs approach outperforms the HMM approach by a large margin

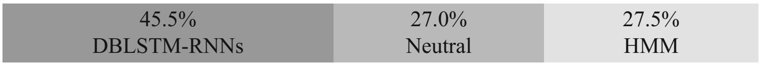


Fig. 9 The percentage preference of the deep BLSTM-RNNS-based and HMM-based video-realistic talking heads

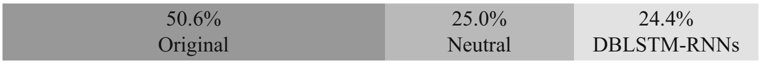


Fig. 10 The percentage preference of the deep BLSTM-RNNS-based and original talking heads

in terms of the two objective metrics. Please note that the computational cost of training DBLSTM-RNN-based talking head is much higher than that of HMM-based one.

7.4 Subjective evaluation

A collection of 10 utterances was chosen from the test set for subjective evaluation of the trajectory tiling approaches. According to the trajectory tiling method described in Section 6, text-driven talking head animation videos were generated for both FBB128 and HMM. For each test utterance, the two talking head videos were placed side-by-side randomly and played with original speech. A group of 20 subjects were asked to perform an A/B preference test according to the naturalness. The percentage preference is shown in Fig. 9. We can clearly see that the DBLSTM-RNN-based talking head is significantly preferred as compared with the HMM-based one. Most subjects prefer the DBLSTM-RNN-based talking head because its lip movement is more smooth and accurate than the HMM-based one. Using the same evaluation method, we also compared the DBLSTM-RNN-based talking head with the original video. The percentage preference is shown in Fig. 10. Results show that about half the votes are given to the origninal vidoes and another half of the votes go to Neutral and BLSTM-RNN. This means, the synthesized videos look quite realsitic and some of them cannnot be distinguished with the original ones. Feedback from subjects shows that the syntheiszed mouth movments look quite smooth and synchronized with speech very well, but in some synthesized vidoes, the mouth openning is smaller as compared with the orignial ones. This is mainly because the generated trajectory is kind of over-smoothed. We believe that a post-filtering method, e.g. global variance (GV) [36], can alleviate this problem. Some video clips of the synthesized taking head can be found from [14].

8 Conclusions

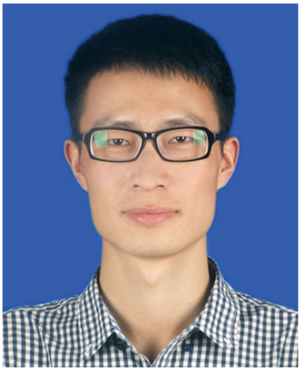
In this paper, we propose a long short-term memory recurrent neural network (LSTM-RNN) approach for video-realistic talking head. At the very beginning, an audio/visual database of a subject talking to a camera with frontal view is required as the training data. The audio/visual stereo data are converted into two parallel temporal sequences, i.e., input sequences and output sequences. Then we use DBLSTM-RNNs to model the temporal and long-range dependencies of these two sequences. The trained DBLSTM-RNNs is used to generate the output parameter sequence from a given input sequence as a guide to select, from the original training database, an optimal sequence of facial images. The face images of low faces are finally stitched back to a background face video of the subject, resulting in video-realistic performance. Our study shows that proposed LSTM-RNN approach significantly outperforms the state-of-the-art HMM-based approach. In future work, we plan to add emotion to this framework to make the talking head more lifelike. A recent study has shown that stack bottleneck DNN features [44] provide another promising way to address the contextual information. We plan to compare the DBLSTM-RNN approach with the stack bottleneck approach in the future work.

Acknowledgments This work was supported by the National Natural Science Foundation of China (Grant No. 61175018 and 61571363).

References

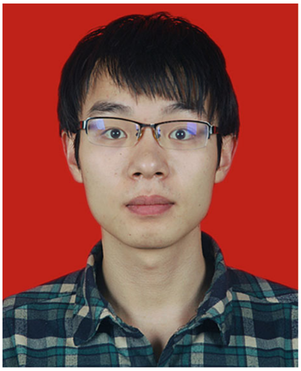
1. Blanz V, Vetter T (1999) A morphable model for the synthesis of 3d faces. In: Siggraph, pp 187–194
2. Blanz V, Basso C, Poggio T, Vetter T (2003) Reanimating faces in images and video. In: Eurographics, pp 641–650
3. Bregler C, Covell M, Slaney M (2007) Video rewrite: driving visual speech with audio. In: Siggraph, pp 353–360
4. Collobert R, Weston J (2008) A unified architecture for natural language processing: deep neural networks with multitask learning. In: ICML, pp 160–167
5. Cootes TF, Edwards GJ, Taylor CJ (2001) Active appearance models. IEEE Trans Pattern Anal Mach Intell 23(6):681–685
6. Cosatto E, Graf HP (2000) Photo-realistic talking heads from image samples. IEEE Trans Multimedia 2(3):152–163
7. Cosatto E, Ostermann J, Graf HP, Schroeter J (2003) Lifelike talking faces for interactive services. Proc IEEE 91(9):1406–1428
8. Ezzat T, Poggio T (2000) Visual speech synthesis by morphing visemes. Int J Comput Vis 38(1):45–57
9. Ezzat T, Geiger G, Poggio T (2002) Trainable videorealistic speech animation. In: Siggraph, pp 388–397
10. Fan Y, Qian Y, Xie F, Soong FK (2014) TTS synthesis with bidirectional LSTM based recurrent neural networks. In: Interspeech, pp 1964–196
11. Graves A (2012) Supervised sequence labelling with recurrent neural networks, In: Springer, p 385
12. Graves A, Mohamed A, Hinton G (2013) Speech recognition with deep recurrent neural networks. In: ICASSP, pp 6645–6649
13. http://marketplace.xbox.com/en-US/Product/Avatar-Kinect/66acd000-77fe-1000-9115-d8025848081a (accessed on 13 June 2015)
14. http://www.nwpu-aslp.org/lips2008 (accessed on 13 June 2015)
15. Hinton G, Deng L, Yu D, Dahl G, Mohamed A, Jaitly N, Senior A, Vanhoucke V, Nguyen P, Sainath T, Kingsbury B (2012) Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. IEEESignal Proc Mag 29(6):82–97
16. Hochreiter S (1998) The vanishing gradient problem during learning recurrent neural nets and problem solutions. Int J Uncertain Fuzziness Knowl-Based Syst 6(02):107–116
17. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
18. Hofer G, Yamagishi J, Shimodaira H (2008) Speech-driven lip motion generation with a trajectory hmm. In: Proceedings of interspeech
19. Jia J, Zhang S, Meng F, Wang Y, Cai L (2011) Emotional audio-visual speech synthesis based on pad. EURASIP J Audio Speech Music Process 19(3):570–582
20. Jia J, Wu Z, Zhang S, Meng H, Cai L (2013) Head and facial gestures synthesis using pad model for an expressive talking avatar. Multimed Tools Appl. doi:10.1007/S11042-013-1604-8
21. Kang S, Qian X, Meng HY Multi-distribution deep belief network for speech synthesis. In: ICASSP, pp 8012–8016
22. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp 1097–1105
23. Leggetter CJ, Woodland PC (1995) Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models, vol 9, pp 171–185
24. Li B, Xie L, Zhou X, Zhang Y (2011) Real-time speech driven talking avatar. J Tsinghua Univ Sci Tech 51(9):1180–1186
25. Liu K, Ostermann J (2009) Optimization of an image-based talking head system. In: EURASIP journal on audio, speech, and music processing, vol 2009
26. Meng F, Wu Z, Jia J, Meng H, Cai L (2013) Synthesizing english emphatic speech for multimodal corrective feedback in computer-aided pronunciation training. Multimed Tools Appl. doi:10.1007/s11042-013-1601-y
27. Mike S, Paliwal K (1997) Bidirectional recurrent neural networks. IEEE Trans Signal Process 45(6):2673–2681
28. Ohman T, Salvi G (1999) Using hmms and anns formapping acoustic to visual speech. TMH-QPSR 40(1):45–50
29. Pérez P, Gangnet M, Blake A (2003) Poisson image editing. ACM Trans Graph 22(3):313–318
30. Pighin F, Hecker J, Lischinski J, Lischinski D, Szeliski D, Salesinet R (2006) Synthesizing realistic facial expressions from photographs. In: Siggraph, p 19

31. Qian Y, Fan Y (2014) On the training aspects of deep neural network (DNN) for parametric TTS synthesis. In: ICASSP, Soong FK, pp 3829–3833
32. Roweis S (1998) EM algorithms for PCA and SPCA. In: Advances in neural information processing systems, pp 626–632
33. Sako S, Tokuda K, Masuko T, Kobayashi T, Kitamura T (2000) HMM-based text-to-audio-visual speech synthesis. In: In ICSLP, pp 25–28
34. Salvi G, Beskow J, Moubayed SA, Granstrom B (2009) Synface: speech-driven facial animation for virtual speech-reading support. In: EURASIP journal on Audio, speech, andmusic processing, vol 2009
35. Theobald BJ, Fagel S, Bailly G (2008) LIPS2008: Visual Speech Synthesis Challenge. In: Interspeech, pp 2310–2313
36. Tomoki T, Tokuda K (2007) A speech parameter generation algorithm considering global variance for HMM-based speech synthesis. IEICE Trans Inf Syst 90(5):816–824
37. Wang L, Soong FK (2014) HMM trajectory-guided sample selection for photo-realistic talking head. In: Multimedia Tools and Applications, pp 1–21
38. Wang Q, Zhang W, Tang X, Shum HY (2006) Real-time bayesian 3-d pose tracking. IEEE Trans Circ Syst Video Tech 16(12):1533–1541
39. Wang L, Qian X, Han W, Soong FK (2010) Synthesizing photo-real talking head via trajectoryguided sample selection. In: Interspeech, pp 446–449
40. Wang L, Han W, Soong FK, Huo Q (2011) Text driven 3d photo-realistic talking head. In: Interspeech, pp 3307–3310
41. Wang L, Qian Y, Scott MR, Chen G, Soong FK (2012) Computer-assisted audiovisual language learning. Computer 45(6):38–47
42. Williams RJ, Zipser D (1989) A learning algorithm for continually running fully recurrent neural networks. Neural Comput 1(2):270–280
43. Wu Z, Zhang S, Cai L, Meng H (2006) Real-time synthesis of chinese visual speech and facial expressions using mpeg-4 fap features in a three-dimensional avatar. In: Proc. Interspeech, pp 1802–1805
44. Wu Z, Valentini-Botinhao C, Watts O, King S (2015) Deep neural networks employing multi-task learining and stacked bottleneck features for speech synthesis. In: ICASSP, pp 4460–4464
45. Xie L, Liu Z (2006) Speech animation using coupled hidden Markov models. In: ICPR, pp 1128–1131
46. Xie L, Liu Z-Q (2007) Realistic mouth-synching for speech-driven talking face using articulatory modelling. IEEE Trans Multimed 9(23):500–510
47. Xie L, Sun N, Fan B (2014) A statistical parametric approach to video-realistic text-driven talking avatar. Multimedia Tool Appl 73(1):377–396
48. Zen H, Tokuda K, Black A (2009) Statistical parametric speech synthesis. Speech Comm 51(11):1039–1064
49. Zen H, Senior A, Schuster M (2013) Statistical parametric speech synthesis using deep neural networks. In: ICASSP, pp 7962–7966

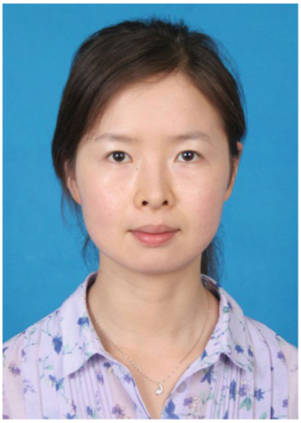


Bo Fan is currently a master student in the School of Computer Science, Northwestern Polytechnical University, Xi'an, China. From 2014 to 2015, he was with the speech group of Microsoft Research Asia, Beijing, China, as an intern student. From 2015 to now, he was a Research Assistant in the Temasek Laboratories@NTU (TL@NTU), NANYANG Technological University. He has published one paper in ICASSP 2015. His current research interests include talking avatar animation, speech synthesis, voice conversion and pattern recognition.

Lei Xie received the Ph.D. degree in computer science from Northwestern Polytechnical University, Xi'an, China, in 2004. He is currently a Professor with School of Computer Science, Northwestern Polytechnical University, Xi'an, China. From 2001 to 2002, he was with the Department of Electronics and Information Processing, Vrije Universiteit Brussel (VUB), Brussels, Belgium, as a Visiting Scientist. From 2004 to 2006, he was a Senior Research Associate in the Center for Media Technology (RCMT), School of Creative Media, City University of Hong Kong, Hong Kong. From 2006 to 2007, he was a Postdoctoral Fellow in the Human-Computer Communications Laboratory (HCCL), Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong. He has published more than 80 papers in major journals and conference proceedings, such as the IEEE TRANSACTIONS ON AUDIO, SPEECH AND LANGUAGE PROCESSING, IEEE TRANSACTIONS ON MULTIMEDIA, INFORMATION SCIENCES, PATTERN RECOGNITION, ACM/Springer Multimedia Systems, ACL, Interspeech, ICPR, ICME and ICASSP. He has served as program chair and organizing chair in several conferences. He is a Senior Member of IEEE, a member of ISCA, a member of ACM, a member of APSIPA and a senior member of China Computer Federation (CCF). He is a Board-of-Governor of the Chinese Information Processing Society of China (CIPSC), a board member of the APSIPA Speech, Language and Audio (SLA) technical committee, a board member of the multimedia technical committee of CCF, a board member of the multimedia technical committee of China Society of Image and Graphics (CSIG). His current research interests include speech and language processing, multimedia and human-computer interaction.



Shan Yang is currently a Ph.D student in the School of Computer Science, Northwestern Polytechnical University, Xi'an, China. His current research interests include talking avatar animation, speech synthesis.



Lijuan Wang received B.E. from Huazhong Univ. of Science and Technology and Ph.D. from Tsinghua Univ., China in 2001 and 2006 respectively. In 2006, she joined the speech group of Microsoft Research Asia, where she is currently a Lead Researcher. Her research areas include data analytics, deep learning (feedforward and recurrent neural networks), and machine learning. She has been the key contributor in developing technologies on printed/handwritten text recognition, avatar animation (talking head), and speech synthesis (TTS)/recognition, which have been shipped into Microsoft products, such as OneNote, Bing dictionary, Tellme, Office Communication, Exchange, etc. She has published more than 25 papers on top conferences and journals and she is the inventor/co-inventor of more than 10 granted/pending USA patents. She is a senior member of IEEE and a member of ISCA.

Frank Soong is a Principal Researcher and Manager of the Speech Group, where speech modeling, recognition, synthesis research is conducted. He received his BS, MS and Ph. D, all in EE from the National Taiwan University, the University of Rhode Island and Stanford University, respectively. He joined Bell Labs Research, Murray Hill, NJ, USA in 1982, worked there for 20 years and retired as a Distinguished Member of Technical Staff in 2001. In Bell Labs, he had worked on various aspects of acoustics and speech processing, including: speech coding, speech and speaker recognition, stochastic modeling of speech signals, efficient search algorithms, discriminative training, dereverberation of audio and speech signals, microphone array processing, acoustic echo cancellation, hands-free noisy speech recognition. He was also responsible for transferring recognition technology from research to AT&T voice-activated cell phones which were rated by the Mobile Office Magazine as the best among competing products evaluated. He was the co-recipient of the Bell Labs President Gold Award for developing the Bell Labs Automatic Speech Recognition (BLASR) software package. He visited Japan twice as a visiting researcher: first from 1987 to 1988, to the NTT Electro-Communication Labs, Musashino, Tokyo; then from 2002–2004, to the Spoken Language Translation Labs, ATR, Kyoto. In 2004, he joined Microsoft Research Asia (MSRA), Beijing, China to lead the Speech Research Group. He is a visiting professor of the Chinese University of Hong Kong (CUHK) and the co-director of CUHKMSRA Joint Research Lab, recently promoted to a National Key Lab of Ministry of Education, China. He was the co-chair of the 1991 IEEE International Arden House Speech Recognition Workshop. He is a committee member of the IEEE Speech and Language Processing Technical Committee of the Signal Processing Society and has served as an associate editor of the Transactions of Speech and Audio Processing. He published extensively and coauthored more than 200 technical papers in the speech and signal processing fields. He is an IEEE Fellow.