PHOTO-REAL TALKING HEAD WITH DEEP BIDIRECTIONAL LSTM

Bo Fan[1,2*], *Lijuan Wang*[2], *Frank K. Soong*[2], *Lei Xie*[1]

[1] School of Computer Science, Northwestern Polytechnical University, Xi'an, China
[2] Microsoft Research Asia, Beijing, China

{bofan,lxie}@nwpu-aslp.org, {lijuanw,frankkps}@microsoft.com

ABSTRACT

Long short-term memory (LSTM) is a specific recurrent neural network (RNN) architecture that is designed to model temporal sequences and their long-range dependencies more accurately than conventional RNNs. In this paper, we propose to use deep bidirectional LSTM (BLSTM) for audio/visual modeling in our photo-real talking head system. An audio/visual database of a subject's talking is firstly recorded as our training data. The audio/visual stereo data are converted into two parallel temporal sequences, i.e., contextual label sequences obtained by forced aligning audio against text, and visual feature sequences by applying active-appearance-model (AAM) on the lower face region among all the training image samples. The deep BLSTM is then trained to learn the regression model by minimizing the sum of square error (SSE) of predicting visual sequence from label sequence. After testing different network topologies, we interestingly found the best network is two BLSTM layers sitting on top of one feed-forward layer on our datasets. Compared with our previous HMM-based system, the newly proposed deep BLSTM-based one is better on both objective measurement and subjective A/B test.

Index Terms— BLSTM, RNN, AAM, talking head

1. INTRODUCTION

Talking heads are useful in applications of human-machine interaction, e.g., reading emails, news or eBooks, acting as an intelligent voice agent or a computer assisted language teacher, etc. A lively, lip-sync talking head can attract the attention of a user, make the human/machine interface more engaging or add entertainment ingredients to an application. Our motivation is to build a photo-real talking head where the animation is video realistic: that is, we desire our talking head to look as much as possible as if it were a camera recording of a human subject, and not that of a cartoon character.

To synthesize visual speech animations from audio-video parallel data, various approaches have been proposed, such as: key-frame based interpolation [1], unit selection synthesis [2], 3D model-based animation [3], HMM-based synthesis and its variants [4–7], and the hybrid approach [8] using the HMM predicted trajectory [9] to guide the sample selection. For both HMM-based parametric and HMM-guided hybrid approaches, the statistically trained HMM is crucial since the HMM predicted visual trajectories to a large extent determine how well the visual lips can be rendered. Although HMM can model sequential data efficiently, there are still some limitations, 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 [10], NN-based approaches have been explored [11] in the speech synthesis field. 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.

There are two mainstream neural net architectures, feed forward vs. recurrent. Recurrent neural networks (RNNs) are able to incorporate contextual information from past inputs, which allows them to instantiate a wide range of sequence-to-sequence maps. Schuster et al. [12] propose the bidirectional RNNs (BRNNs) which can incorporate contextual information from both past and future inputs. But conventional RNNs cannot well model the long-span relations in sequential data because of the vanishing gradient problem. Hochreiter et al. [13] found that the LSTM architecture, which uses purpose-built memory cells to store information, is better at finding and exploiting long range context. Combining BRNNs with LSTM gives BLSTM, which can access long-range context in both directions.

In this paper, we propose a deep BLSTM-based approach for visual speech synthesis. The audio/visual parallel training data are converted into sequences of contextual labels and visual feature vectors, respectively. Like [14], we adopt the AAM algorithm to model the lower face and take the low dimensional appearance parameters as the visual features. The deep BLSTM neural network is trained to learn the regression model between the two audio and visual parallel sequences by minimizing the generation errors, in which the input layer is the label sequence and the output prediction layer is the AAM visual parameter sequence. In the synthesis stage, the predicted AAM visual parameter sequence can be restored back to high quality photo realistic face images and render the full face talking head with lip-synced animation.

The rest of this paper is organized as follows. Section 2 gives an overview of the system. The audio/visual feature representation and extraction are described in Section 3. Section 4 presents the deep BLSTM architecture and training. Experimental results are discussed in Section 5. Finally, we draw our conclusions in Section 6.

2. SYSTEM OVERVIEW

Fig. 1 shows the system overview of the proposed photo-real talking head using deep BLSTM networks. Firstly, an audio/visual database of a subject talking to a camera with frontal view of his/her face is recorded as our training data. In the training stage, the audio

*This work has been done when the first author visited Microsoft Research Asia as a summer intern. This work was partially supported by a grant from the National Natural Science Foundation of China (61175018) and a grant from the Fok Ying Tung Education Foundation (131059).

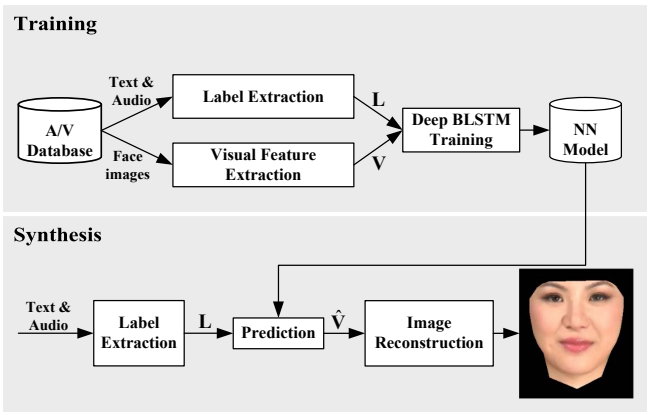


Fig. 1. System overview of the proposed talking head.

is converted into a sequence of contextual phoneme labels $\mathbf{L}$ using forced alignment, and the corresponding lower face image sequence is transformed into AAM feature vectors $\mathbf{V}$. Then we train the deep BLSTM neural networks to learn the regression model between the two audio and visual parallel sequences by minimizing the SSE of the prediction, in which the input layer is the label sequence $\mathbf{L}$ and the output prediction layer is the visual feature sequence $\mathbf{V}$. In the synthesis stage, for any input text with natural or synthesized speech by text-to-speech (TTS), we first extract the label sequence $\mathbf{L}$ from them and then predict the visual AAM parameters $\hat{\mathbf{V}}$ using the well trained deep BLSTM network. Finally, the predicted AAM visual parameter sequence $\hat{\mathbf{V}}$ can be reconstructed to high quality photo realistic face images and rendering the full face talking head with lip-synced animation.

3. AUDIO/VISUAL FEATURE REPRESENTATION

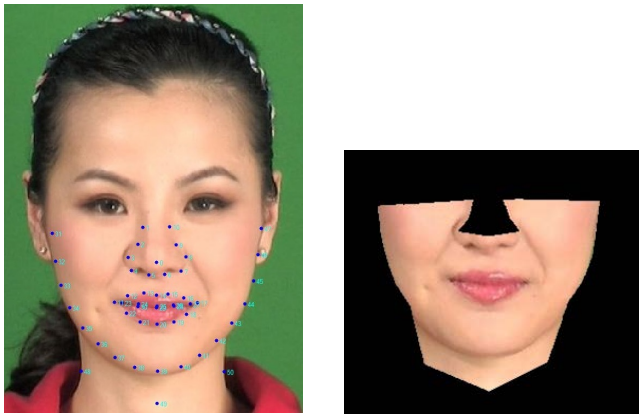
3.1. Contextual labels L

The input of a desired talking head system can be any arbitrary text along with natural audio recordings or synthesized speech by TTS. For natural recordings, the phoneme/state time alignment can be obtained by conducting forced alignment using a trained speech recognition model. For TTS synthesized speech, the phoneme/state sequence and time offset are a by-product of the synthesis process. Therefore, for each speech utterance, 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. 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 natural recordings or TTS synthesized speech. Please note that the contextual label can be easily extended to contain richer information, like positions in syllables, in 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.



(a) 51 facial feature points. (b) The texture of a lower face.

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

3.2. AAM visual feature V

In our system, the visual stream is a sequence of lower face images which are strongly correlated to the underlying speech. As the raw face image is hard to model directly due to the high dimensionality, we use AAM [15] for visual feature extraction. AAM is a joint statistical model compactly representing both the shape and the texture variations and the correlation between them.

Since the speaker moves his/her head naturally during recording, 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 [16], 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:

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

where $j = 1, 2, \ldots, J$ and J is the total number of the face images. In this work, we use a set of 51 facial feature points, as shown in Fig. 2 (a). The mean shape is simply defined by

$$\mathbf{s}_0 = \sum_{j=1}^{J} \mathbf{s}_j / J. \quad (2)$$

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

$$\mathbf{s}_j = \mathbf{s}_0 + \sum_{i=1}^{N_{\text{shape}}} a_{ji}\tilde{\mathbf{s}}_i = \mathbf{s}_0 + \mathbf{a}_j\mathbf{P}_s, \quad (3)$$

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.

Accordingly, the texture of the j-th face image, $\mathbf{t}_j$, is defined by a vector concatenating the R/G/B value of every pixel that lies inside the mean shape $\mathbf{s}_0$:

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

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

As the dimensionality of the texture vector is too high to use PCA directly, we apply EMPCA [17] to all J textures. As a result, the j-th texture $\mathbf{t}_j$ can be given approximately by:

$$\mathbf{t}_j = \mathbf{t}_0 + \sum_{i=1}^{N_{\text{texture}}} b_{ji}\tilde{\mathbf{t}}_i = \mathbf{t}_0 + \mathbf{b}_j\mathbf{P}_t, \quad (5)$$

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

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:

$$(\mathbf{a}_j, \mathbf{b}_j) = \sum_{i=1}^{N_{\text{appearance}}} v_{ji}\tilde{\mathbf{v}}_i = \mathbf{v}_j\mathbf{P}_v, \tag{6}$$

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

$$\mathbf{s}_j = \mathbf{s}_0 + \mathbf{v}_j\mathbf{P}_{vs}\mathbf{P}_s = \mathbf{s}_0 + \mathbf{v}_j\mathbf{Q}_s, \tag{7}$$

$$\mathbf{t}_j = \mathbf{t}_0 + \mathbf{v}_j\mathbf{P}_{vt}\mathbf{P}_t = \mathbf{t}_0 + \mathbf{v}_j\mathbf{Q}_t. \tag{8}$$

Now, we can reconstruct the shape and texture of the j-th lower face image by only one parameter vector $\mathbf{v}_j$, and $\mathbf{v}_j$ is the j-th appearance parameter vector which we use as AAM visual feature. Subsequently, the lower face sequence with T frames can be represented by the visual feature sequence $\mathbf{V} = (\mathbf{v}_1, \ldots, \mathbf{v}_t, \ldots, \mathbf{v}_T)$.

4. DEEP BLSTM FOR TALKING HEAD ANIMATION

4.1. Network structure

The extracted label sequence $\mathbf{L}$ and visual feature sequence $\mathbf{V}$ are two time varying parallel sequences. After resampling, we can easily make the two sequences the same frame rate. In our BLSTM network, as shown in Fig. 3, label sequence $\mathbf{L}$ is the input layer, and visual feature sequence $\mathbf{V}$ serves as the output layer and H denotes the hidden layer. In particular, at t-th frame, the input of the network is the t-th label vector $\mathbf{l}_t$ and the output is the t-th visual feature vector $\mathbf{v}_t$. As described in [18], 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. 3, more hidden layers can be added in to construct a deep BLSTM.

In the training stage, we have multiple sequence pairs of $\mathbf{L}$ and $\mathbf{V}$. As we represent both sequences as continuous numerical vectors, the network is treated as a regression model minimizing the SSE of predicting $\hat{\mathbf{V}}$ from $\mathbf{L}$. In the test (or synthesis) stage, given any arbitrary text along with natural or synthesized speech, we firstly convert them into a sequence of labels, then feed into the trained BLSTM network, and the output of the network is the predicted visual AAM feature sequence. After reconstructing the AAM feature vectors to RGB images, we can get the photo 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.

4.2. Network training

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

$$E(\mathbf{w}) = \sum_{k=1}^{M_{\text{train}}} E_k(\mathbf{w}), \tag{9}$$

where M_{train} represents the number of sequences in the training data and $\mathbf{w}$ denotes the network inter-node weights. In our task, the training criterion is to minimize the SSE between the predicted visual features $\hat{\mathbf{V}} = (\hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2, \ldots, \hat{\mathbf{v}}_T)$ and the ground truth $\mathbf{V} = (\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_T)$. For a particular input sequence k, the error function takes the form

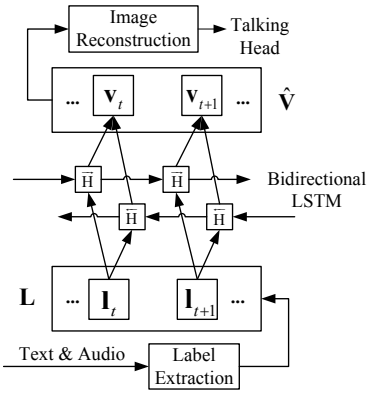


Fig. 3. BLSTM neural network in our talking head system.

$$E_k(\mathbf{w}) = \sum_{t=1}^{T_k} E_{kt} = \frac{1}{2}\sum_{t=1}^{T_k} \left\| \hat{\mathbf{v}}_t^k - \mathbf{v}_t^k \right\|^2, \tag{10}$$

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:

$$\Delta\mathbf{w}(r) = m\Delta\mathbf{w}(r-1) - \alpha\frac{\partial E(\mathbf{w}(r))}{\partial \mathbf{w}(r)}, \tag{11}$$

where $0 \le \alpha \le 1$ is the learning rate, $0 \le m \le 1$ 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 [19, 20] 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. Taking the error function derivatives with respect to the output of the network as an example. For $\hat{\mathbf{v}}_t^k = (\hat{v}_{t1}^k, \ldots, \hat{v}_{tj}^k, \ldots, \hat{v}_{tN_{\text{appearance}}}^k)$ in k-th $\hat{\mathbf{V}}$, because the activation function used in the output layer is an identity function, we have

$$\hat{v}_{tj}^k = \sum_h w_{oh} z_{ht}^k, \tag{12}$$

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

$$\frac{\partial E_{kt}}{\partial w_{oh}} = \sum_{j=1}^{N_{\text{appearance}}} \frac{\partial E_{kt}}{\partial \hat{v}_{tj}^k}\frac{\partial \hat{v}_{tj}^k}{\partial w_{oh}}, \tag{13}$$

and according to Eq. (10) and (12), we can derive

$$\frac{\partial E_{kt}}{\partial w_{oh}} = \sum_{j=1}^{N_{\text{appearance}}} (\hat{v}_{tj}^k - v_{tj}^k) z_{ht}^k, \tag{14}$$

$$\frac{\partial E_k}{\partial w_{oh}} = \sum_{t=1}^{T} \frac{\partial E_{kt}}{\partial w_{oh}}. \tag{15}$$

5. EXPERIMENTS

5.1. Experimental setup

Our experiments were carried out on the A/V database with 593 English utterances spoken by a female in a neutral style. The transcription is from ARCTIC-part A [21], which is designed for good phonetic coverage and contextual diversity. The frame rate of the video

Table 1. *The objective experimental results for networks with different hidden layers and numbers of nodes.*

Node	128				256				512			
TP	RMSE (shape)	RMSE (texture)	RMSE (appearance)	CORR	RMSE (shape)	RMSE (texture)	RMSE (appearance)	CORR	RMSE (shape)	RMSE (texture)	RMSE (appearance)	CORR
BBB	1.133	6.307	157.271	0.642	1.146	6.393	160.279	0.625	1.158	6.411	161.024	0.621
BBF	1.123	6.309	157.378	0.643	1.725	7.902	213.273	0.002	1.726	7.904	213.333	-0.016
BFB	1.142	6.376	159.747	0.630	1.155	6.433	161.658	0.622	1.147	6.380	159.827	0.632
BFF	1.151	6.379	159.773	0.631	1.148	6.427	161.544	0.626	1.726	7.903	213.295	-0.009
FBB	**1.122**	**6.286**	**156.502**	**0.647**	**1.137**	**6.320**	**157.725**	**0.641**	**1.133**	**6.314**	**157.549**	**0.643**
FBF	1.129	6.327	158.061	0.640	1.726	7.903	213.312	-0.021	1.725	7.899	213.153	0.032
FFB	1.148	6.380	160.007	0.630	1.141	6.385	158.682	0.638	1.528	7.666	204.823	0.185
FFF	1.726	7.905	213.354	-0.032	1.394	7.124	186.012	0.430	1.726	7.909	213.486	-0.014

files is 25 fps and all together 81974 face images with pixel resolution 720×576 are collected. We divided the database into 3 parts randomly, 80% for training, 10% for validation and 10% for testing. We randomly selected 20000 images from the training set for lower-face AAM training. We chose top 66 shape and 100 texture principal components containing about 99% and 87% cumulative energy contents, respectively. The final dimension of the visual appearance vector ($\mathbf{v}_t^k$) is 87. We found 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.

We conducted objective evaluations by directly comparing the predicted visual AAM features with the ground truth. Four objective metrics are used, defined as follows [8]:

$$RMSE(shape) = \frac{\sum_{k=1}^{M_{\text{test}}} \sum_{t=1}^{T_k} \sqrt{\left\| \hat{\mathbf{s}}_t^k - \mathbf{s}_t^k \right\|^2 / N_{\text{shape}}}}{\sum_{k=1}^{M_{\text{test}}} T_k}, \quad (16)$$

$$RMSE(texture) = \frac{\sum_{k=1}^{M_{\text{test}}} \sum_{t=1}^{T_k} \sqrt{\left\| \hat{\mathbf{t}}_t^k - \mathbf{t}_t^k \right\|^2 / N_{\text{texture}}}}{\sum_{k=1}^{M_{\text{test}}} T_k}, \quad (17)$$

$$RMSE(appearace) = \frac{\sum_{k=1}^{M_{\text{test}}} \sum_{t=1}^{T_k} \sqrt{\left\| \hat{\mathbf{v}}_t^k - \mathbf{v}_t^k \right\|^2 / N_{\text{appearance}}}}{\sum_{k=1}^{M_{\text{test}}} T_k}, \quad (18)$$

$$CORR = \frac{\sum_{k=1}^{M_{\text{test}}} \sum_{t=1}^{T_k} corr(\hat{\mathbf{v}}_t^k, \mathbf{v}_t^k)}{\sum_{k=1}^{M_{\text{test}}} T_k}, \quad (19)$$

where $corr(\hat{\mathbf{v}}_t^k, \mathbf{v}_t^k)$ denotes the correlation coefficient and $\hat{\mathbf{s}}_t^k$ and $\hat{\mathbf{t}}_t^k$ are shape/texture parameters reconstructed from $\hat{\mathbf{v}}_t^k$. Note that lower RMSE and higher CORR correspond to better performance.

5.2. Different network topologies

We tested the performance of network topologies with different hidden layers (F–feed forward, B–BLSTM) and numbers of nodes, as described in Table 2. Results show that the 3-hidden-layer structures outperform the 1- and 2-hidden layer structures in general. The results for all tested 3-hidden-layer structures are summarized in Table 1. We interestingly found that, in terms of the four objective metrics, the best network topology is two BLSTM layers sitting on top of one feed-forward layer (FBB) and FBB with 128 nodes per layer obtains the best performance.

5.3. Deep BLSTM vs. HMM

We also compared our deep BSLTM approach with our previous HMM-based approach [8]. In the HMM-based system, five-state, left-to-right HMM phone 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. The results for FBB128 deep BLSTM and HMM are shown in Table 3. We can clearly see that the deep BLSTM approach outperforms the HMM approach by a large margin in terms of the 4 objective metrics. Please note that the computational cost of training BLSTM-based talking head is much higher that of HMM-based one.

Table 2. Network topologies tested in our experiments.

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

Table 3. *Comparison between deep BLSTM and HMM.*

Comparison	RMSE (shape)	RMSE (texture)	RMSE (appearance)	CORR
HMM	1.223	6.602	167.540	0.582
deep BLSTM	**1.122**	**6.286**	**156.502**	**0.647**

5.4. Subjective evaluation

For subjective evaluation, we chose 10 sequences of labels randomly from the test set and rendered the deep BLSTM-based and the HMM-based talking head videos, respectively. For each test sequence, the two talking head videos were played side-by-side randomly 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. 4. We can clearly see that the deep BLSTM-based talking head is significantly preferred to the HMM-based one. Most subjects prefer the BLSTM-based talking head because its lip movement is more smooth than the HMM-based one. Some video clips of the synthesized taking head can be found from [22].

Fig. 4. The percentage preference of the deep BLSTM-based and HMM-based photo-real talking heads.

6. CONCLUSION

In this paper, we propose to use deep BLSTM to model the temporal and long-range dependencies of audio/visual stereo data for a photo-real talking head animation. Our study shows that the best network is two BLSTM layers sitting on top of one feed-forward layer on our datasets. Compared with our previous HMM-based approach, the proposed deep BLSTM shows superior performances in both objective and subjective tests. In future work, we plan to take richer information into account for the contextual label (besides simple phonetic information). We believe the deep BSLTM approach is promising in achieving expressiveness in talking head animation.

7. REFERENCES

[1] C. Bregler, M. Covell, and M. Slaney, "Video rewrite: driving visual speech with audio," in *Proc. ACM SIGGGRAPH 97*, 1997, pp. 353–360.

[2] K. Liu and J. Ostermann, "Realistic facial animation system for interactive services," in *Proc. Interspeech*, 2008, pp. 2330–2333.

[3] L.J. Wang, W. Han, and F. Soong, "High quality lip-sync animation for 3D photo-realistic talking head," in *Proc. ICASSP*. IEEE, 2012, pp. 4529–4532.

[4] S. Sako, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "HMM-based text-to-audio-visual speech synthesis.," in *Proc. Interspeech*, 2000, pp. 25–28.

[5] S. Fu, R. Gutierrez-Osuna, A. Esposito, P. Kakumanu, and O. Garcia, "Audio/visual mapping with cross-modal hidden Markov models," *IEEE Transactions on Multimedia*, vol. 7, no. 2, pp. 243–252, 2005.

[6] L. Xie and Z.Q. Liu, "Realistic mouth-synching for speech-driven talking face using articulatory modelling," *IEEE Transactions on Multimedia*, vol. 9, no. 3, pp. 500–510, 2007.

[7] L. Xie and Z.Q. Liu, "A coupled HMM approach to video-realistic speech animation," *Pattern Recognition*, vol. 40, no. 8, pp. 2325–2340, 2007.

[8] L.J. Wang and F. Soong, "HMM trajectory-guided sample selection for photo-realistic talking head," *Multimedia Tools and Applications*, pp. 1–21, 2014.

[9] L.J. Wang, Y.J. Wu, X.D. Zhuang, and F. Soong, "Synthesizing visual speech trajectory with minimum generation error," in *Proc. ICASSP*. IEEE, 2011, pp. 4580–4583.

[10] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.

[11] H. Zen, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *Proc. ICASSP*. IEEE, 2013, pp. 7962–7966.

[12] S. Mike and K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 6, pp. 2673–2681, 1997.

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

[14] L. Xie, N.C. Sun, and B. Fan, "A statistical parametric approach to video-realistic text-driven talking avatar," *Multimedia Tools and Applications*, vol. 73, no. 1, pp. 377–396, 2014.

[15] 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.

[16] Q. Wang, W.W. Zhang, X.O. Tang, and H.Y. Shum, "Real-time bayesian 3-d pose tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 12, pp. 1533–1541, 2006.

[17] S. Roweis, "EM algorithms for PCA and SPCA," *Advances in neural information processing systems*, pp. 626–632, 1998.

[18] A. Graves, *Supervised sequence labelling with recurrent neural networks*, vol. 385, Springer, 2012.

[19] R. J. Williams and D. Zipser, "Gradient-based learning algorithms for recurrent networks and their computational complexity," *Back-propagation: Theory, architectures and applications*, pp. 433–486, 1995.

[20] P. J. Werbos, "Consistency of HDP applied to a simple reinforcement learning problem," *Neural networks*, vol. 3, no. 2, pp. 179–189, 1990.

[21] J. Kominek and A. W. Black, "The CMU Arctic speech databases," in *Fifth ISCA Workshop on Speech Synthesis*, 2004.

[22] `http://research.microsoft.com/en-us/projects/blstmtalkinghead/`.