JOINT AUDIO-VIDEO DRIVEN FACIAL ANIMATION

Xin Chen, Chen Cao, Zehao Xue, Wei Chu

Snap Research, Snap Inc.
63 Market St, Venice, CA 90291
{xin.chen, chen.cao, zehao.xue, wei.chu }@snap.com

ABSTRACT

Automatic facial animation is a research topic of broad and current interest with widespread impact on various applications. In this paper, we present a novel joint audio-video driven facial animation system. Unlike traditional methods, we incorporate a large vocabulary continuous speech recognition (LVCSR) system to obtain phoneme alignments. The use of LVCSR reduces the high error rate associated with the traditional phoneme recognizer. We also introduce a knowledge guided 3D blendshapes modeling for each phoneme to avoid collecting training data and introducing bias from computer vision generated targets. To further improve the quality, we adopt video tracking and jointly optimize the facial animation by combining both sources. In the evaluations, we present both objective study and several subjective studies on three settings: audio-driven, video-driven, and joint audio-video driven. We find that the quality of our proposed system's facial animation generation surpasses that of the recent state-of-the-art systems.

Index Terms— large vocabulary continuous speech recognition (LVCSR), phoneme alignment, lip sync, facial animation.

1. INTRODUCTION

Face tracking and performance-based facial animation have been widely studied and have impacted a wide variety of applications such as computer gaming, animations, and human-computer interface. [1] reported that the humans' level of trust can be increased by 30% when humans interact with a talking head, as compared to text-only scripts.

In this paper, we revisit this problem by applying the latest state-of-the-art technologies in order to measure the resultant improvement of quality. We presented a novel joint audio-video driven facial animation system. Unlike the traditional methods, we incorporated a full-scale state-of-the-art large vocabulary continuous speech recognition (LVCSR) with a strong language model for speech recognition, and obtained phoneme alignment from the word lattice. To our knowledge, we are the first group that applies knowledge-guided 3D blendshapes modeling for phonemes, utilizing 3D face scans to avoid collecting training data or introducing bias from computer vision generated blendshapes. Finally, to further improve the quality, we introduce computer vision generated tracking and jointly generate the facial animation by combining audio and video information. In subjective evaluation, we show that our proposed system performs more optimally than audio-only or video-only systems. We also compare our results with [2] and user study shows that 83% of users prefer our generated animation. [1]

This paper is organized according to the following sections. In section 2, we describe the related work on audio and video driven facial animations, and in section 3 we describe the speech recognition module we used for reliable phoneme alignment. Section 4 offers an in-depth description of generating facial animation by jointly optimizing audio and video information. We then describe our experiment set-up and show comprehensive objective and subjective evaluation results in section 5, and conclude our work in section 6.

[1] https://sites.google.com/site/lipsync2018/

2. RELATED WORK

Performance-based facial animation is the most common technique to generate realistic character facial animation for movies and games. Over the past several years, researchers have been actively working in this area to achieve accurate and expressive facial animation results. Some techniques need special equipment such as physical markers, structured light, or camera arrays. Recently, monocular camera based face tracking and animation have been widely explored because they are more practical for everyday users. Using a single depth camera ([3] [4] [5]) or video camera ([6][7]), existing work can achieve impressive facial animation results. However, fast and subtle mouth motions in visual input may result in a loss of lip shape changing. While humans are very sensitive to the detail lip shape changing, it is important to consider it in facial animation.

Audio-driven facial animation has also been deeply explored in the speech and graphics fields [8]. Those methods can be mainly classed into two categories. The first category is direct conversion by mapping raw speech features, such as Mel-Frequency Cepstral Coefficients (MFCC) to visual parameters [9] [10] [11] [12] [13]. This approach normally requires more corresponding audio-video training data for better generalized performance. In another category, the speech is primarily mapped into a phoneme or phoneme state feature and then the phoneme level feature is mapped to the visual parameters [14] [15] [16] [17]. This approach enables the use of widely available speech recognition corpus, but requires additional data and steps for training, which complicates the process.

Jointly Video-Speech driven facial animation combines the advantages of video and audio. The most related recent work on our paper is [2]. This work uses both video and acoustic input in tracking 3D facial motions. It use a real-time speaker independent DNN-based acoustic model to extract Phoneme State Posterior Probabilities (PSPP) as the interval feature for lip motion regressor. Differing from this method, our method incorporates a full scale LVCSR with a strong language model for word lattice generation and phoneme sequence estimation. Unlike the regression model which: 1. requires additional training data; 2. has potential issue from inaccurate auto generated facial regression target parameters, we develop a phonemes-to-knowledge guided 3D blendshapes mapping model. Experiments show that our system can achieve more accurate and expressive facial animation results.

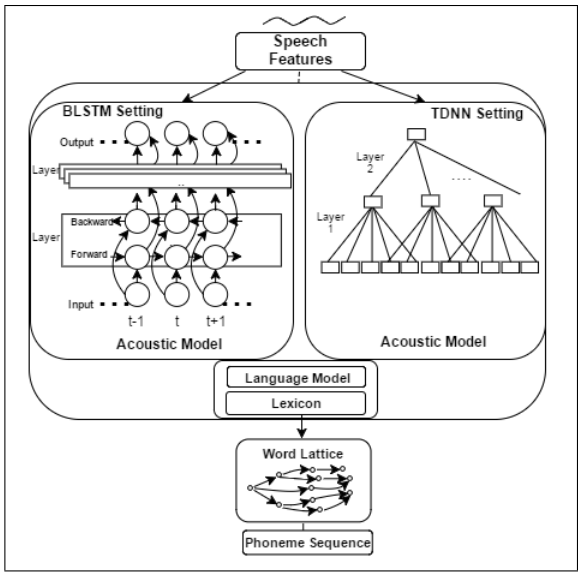


Fig. 1. An illustration of our speech recognition module

3. AUDIO TO BLENDSHAPE MAPPING

In this section, we present our automatic speech recognition system for phoneme recognition, a novel method to create phoneme to blendshapes mapping guided by 3D face scans.

3.1. Large vocabulary continuous speech recognition

In this section, we briefly describe the state-of-the-art speech recognition module that is used in this work.

First, an advanced Bi-directional Long Short-Term Memory (LSTM) [18] or a Time Delayed Neural Network(TDNN) [19] is trained to convert input features to state posteriors. The input features are obtained through passing framed raw audio signals through a Mel-scaled filter bank. I-Vector is also used to improve the performance [20]. Unlike the commonly used cross entropy criterion, the training procedure is a lattice-free version of MMI, in which the denominator state posteriors are obtained by the forward-backward algorithm on an hidden Markov model formed from a phone-level decoding graph. Additionally, the numerator state posteriors are obtained by a similar forward-backward algorithm, but limited to sequences corresponding to the transcript. For each output index of the neural net, a derivative of of the difference between the numerator and denominator occupation probabilities is calculated and propagated back to the network.

In the inferring stage, as illustrated in Fig.1, the trained acoustic model, a decision tree for mapping the states to phones, a lexicon, and a pre-trained N-gram language model are used to generate a weight finite state transducer (WFST) [21]. When feeding input features, the output of the arcs in WFST containing words with likelihoods can be transferred into a lattice. From the lattice, the most likely spoken word sequence is obtained through an BFS. The final phoneme sequence $\{\mathbf{p}^1, \mathbf{p}^2, .., \mathbf{p}^T\}$ with starting and ending time T of all phonemes can be obtained through inferring from the most probable word sequence and lattice.

3.2. Knowledge guided phonemes to blendshapes mapping

In this section, we present our novel approach to map phonemes to expression blendshapes. We first ask an actor to pronounce all 39 phonemes from CMU dictionary [22], and use an off-the-shelf face scan device Bellus3D [23] to scan all the 3D face shapes (Fig.2 top).

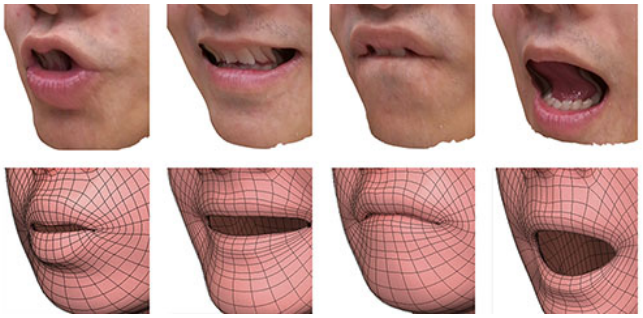


Fig. 2. Four sample phoneme 3D scans and blendshapes generated matching the scans. Phonemes from left to right: R, IY, F, AH.

To represent the face shapes of users' different expressions for facial animation, we adopt the expression blendshape model. Specially, we use the expression description from FaceWarehouse [24], which contains 46 action units as described by Facial Action Coding System (FACS) [25]. We ask an artist to manually create a general expression blendshape model: $\{\mathbf{B}_0^g, \mathbf{B}_1^g, ..., \mathbf{B}_{46}^g\}$, where $\mathbf{B}_0^g$ is the neutral expression shape and $\{\mathbf{B}_1^g, ..., \mathbf{B}_{46}^g\}$ are 46 different expression shapes. Thus any expression's shape $\mathbf{F}^g$ can be represented by a linear combination of blendshape model: $\mathbf{F}^g = \mathbf{B}_0^g + \sum_{i=1}^{46} \beta_i \mathbf{B}_i^g$, where $\mathbf{b} = \{\beta_1, \beta_2, ..., \beta_{46}\}$ is expression coefficient vector.

With the guide of scanned shapes for 39 phonemes, we ask the artist to manually tune the expression coefficients $\mathbf{b}_j^p$ for each phoneme. Bring $\mathbf{b}_j^p$ to the equation above we can generate the face shape matching the scanned shape (Fig.2 bottom). We then concatenate these vectors to build matrix $\mathbf{M}$, where the j-th column is $\mathbf{b}_j^p$.

4. JOINT FACIAL ANIMATION

In this section, we will introduce the pipeline to track 3D face, based on video and audio. We first track 2D facial landmarks, which locate 2D positions of face feature points (Sec 4.1). Based on these tracked 2D landmarks, we reconstruct the 3D face model in Sec.4.2 and track 3D face in Sec.4.3.

4.1. 2D facial landmarks tracking

Facial landmarks correspond to the semantic facial feature positions of human face, such as the eyes corners, lips contour, nose tip etc. In this paper we use off-the-shelf 2D facial landmark tracking algorithm [26], which formulates the facial landmark tracking problem as the optimization of landmarks' error, and learns boosted regression tree to minimize this error. Using this method, we can obtain 68 2D facial landmarks of each video frame in real-time (Fig.3 (a)(c)).

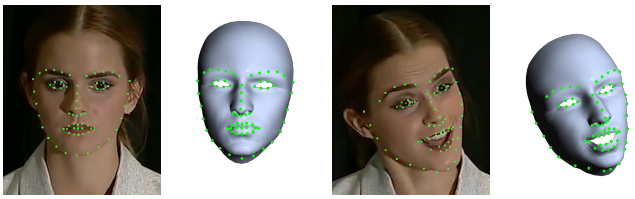
4.2. 3D face modeling

Based on the tracked 2D facial landmarks, we then reconstruct user's 3D face model. We first fit the identity, which describes user's identity shape under neutral expression. With the fitted identity, we then reconstruct user-specific expression blendshapes, which describe the shapes of user's different expressions.

Identity Blanz and Vetter [27] captured and reconstructed 3D shape of 200 people's neutral faces, and performed principal components analysis (PCA) on these shapes' vertex positions. Thus any neutral face shape $\mathbf{F}$ can be represented by a linear combinations of these principal components:

$$\mathbf{F} = \bar{\mathbf{A}} + \sum_{i=1}^{n} \alpha_i \mathbf{A}_i, \quad (1)$$

where $\bar{\mathbf{A}}$ and $\{\mathbf{A}_i\}$ are the mean vector and PCA vectors of morphable model respectively, and $\mathbf{a} = \{\alpha_i\}$ are identity coefficients.



(a) (b) (c) (d)

Fig. 3. Examples of tracked 2D facial landmarks and 3D shape. (a)(b): 2D landmarks of neutral image and modeled 3D face shape; (c)(d): 2D landmarks of expression image and its tracked 3D shape.

Without loss of generality, we assume that user has a neutral expression at the first frame of video (Fig.3(a)), with corresponding 2D facial landmarks vector $\mathbf{P} = \{p_1, p_2, ..., p_{68}\}$. To match the 3D face to these 2D facial landmarks, we first transform the reconstructed 3D face shape from object-coordinate to camera-coordinate by applying a rigid rotation and translation, and then project the transformed face shape into screen coordinate via the camera matrix:

$$\hat{\mathbf{F}} = \mathbf{\Pi}\left(\mathbf{R} \cdot \mathbf{F} + \mathbf{t}\right), \tag{2}$$

where $\hat{\mathbf{F}}$ is the projected face shape, $\mathbf{\Pi}(\cdot)$ is the projection operator using the camera intrinsic matrix, which is defined by the camera, $\mathbf{R}$ and $\mathbf{t}$ are rigid rotation and translation respectively. For more details please refer [6].

To match the projected face shape $\hat{\mathbf{F}}$ with tracked landmarks $\mathbf{P}$, we pre-define the corresponding vertex indices on 3D face shape (Green points in Fig.3 (b)). Similar with [7], we update the mesh vertex indices corresponding to landmarks along the face contour according to current projected face shape $\hat{\mathbf{F}}$. Please refer [7] for more details. With the corresponding vertex indices $\{v_1, v_2, ..., v_{68}\}$, we formulate the error matching 3D face shape to the 2D landmarks as:

$$E^f_{iden} = \sum\nolimits_{k=1}^{68} \left\| \hat{\mathbf{F}}^{(v_k)} - p_k \right\|^2, \tag{3}$$

where $\hat{\mathbf{F}}^{(v_k)}$ is the v_k-th vertex's position of face shape $\hat{\mathbf{F}}$.

We regularize the identity coefficients $\mathbf{a} = \{\alpha_i\}$ based on the estimated probability distribution of 3D morphable model's PCA. Assuming σ_i^2 is the i-th eigenvalue of the face covariance matrix from PCA, let $\Lambda = diag\left(\sigma_1^2, \sigma_2^2, ..., \sigma_n^2\right)$, we define the Tikhonov regularization energy term as:

$$E^r_{iden} = \frac{1}{2}\mathbf{a}^T \Lambda \mathbf{a}. \tag{4}$$

Combining Eqn. 3 and Eqn. 4 we can define the total energy as: $E_{iden} = E^f_{iden} + \omega^r_{iden} E^r_{iden}$, where ω^r_{iden} balances the regularization term and is set as 5.0 in this paper. The unknown parameters here include the rigid rotation $\mathbf{R}$, translation $\mathbf{t}$ and identity coefficients $\mathbf{a} = \{\alpha_i\}$. We use coordinate-descent method to optimize this energy: in each iteration, we only optimize one unknown parameter while fixing others in least-squares way. In our experiment, we find 3 iterations can give a satisfied result. Please notice that in each iteration we need to update the mesh vertex indices corresponding to landmark along face contour. With the fitted identity coefficients $\mathbf{a} = \{\alpha_i\}$, we generate user's neutral face shape $\mathbf{B}_0$ using Eqn.1 (Fig.3(b)).

Expressions We then generate user-specific expression blendshape model. With the general blendshape model $\{\mathbf{B}^g_0, \mathbf{B}^g_1, ..., \mathbf{B}^g_{46}\}$ generated by artist and user's neutral face shape $\mathbf{B}_0$, We use deformation transfer [28] to transfer the 46 expression shapes, result in user-specific blendshape model: $\{\mathbf{B}^g_0, \mathbf{B}^g_1, ..., \mathbf{B}^g_{46}\}$. With this user-specific blendshape model, we can represent user's different expressions by:

$$\mathbf{F} = \mathbf{B}_0 + \sum\nolimits_{i=1}^{46} \beta_i \mathbf{B}_i, \tag{5}$$

where $\mathbf{b} = \{\beta_1, \beta_2, ..., \beta_{46}\}$ are expression coefficients.

4.3. 3D face tracking

With the reconstructed user-specific expression blendshapes, for each input frame at frame t, we can track the 3D face model parameters combining input video and audio. These parameters include: rigid rotation $\mathbf{R}^t$, translation $\mathbf{t}^t$, and non-rigid facial expression coefficients $\mathbf{b}^t = \{\beta^t_1, \beta^t_2, ..., \beta^t_{46}\}$. Next, we will describe the energy terms formulated by these parameters.

Landmark term is used to describe the alignment between the tracked 3D face shape and 2D facial landmarks from Sec.4.1. We first reconstruct the expression face shape using Eqn.5. Similarly to the identity fitting process, we then apply the rigid rotation and translation to the reconstructed face shape, and project the shape into screen coordinate via Eqn.2. With the projected face shape $\hat{\mathbf{F}}$, we can formulate the landmark term as:

$$E^f = \sum\nolimits_{k=1}^{68} \left\| \hat{\mathbf{F}}^{(v_k)} - p_k \right\|^2, \tag{6}$$

where the related parameters have the same meanings with Eqn.3.

Phoneme term is used to describe the alignment between tracked expression coefficients and estimated phonemes from Sec.3. Assume that we have the phoneme vector at frame t as $\mathbf{p}^t$ from Sec.3.1, and the mapping matrix $\mathbf{M}$ from Sec.3.2, we can formulate the phoneme term as:

$$E^p = \left\| \mathbf{b}^t - \mathbf{M}\mathbf{p}^t \right\|^2. \tag{7}$$

Smooth term is used to enhance the smoothness of tracking results. We keep the expression coefficients of previous frame as $\mathbf{b}^{t-1}$, and formulate the smooth term as

$$E^s = \left\| \mathbf{b}^t - \mathbf{b}^{t-1} \right\|^2. \tag{8}$$

Putting these three energy terms together we get the total energy function as $E = E^f + \omega^p E^p + \omega^s E^s$, where ω^p and ω_s balance the different energy terms, which are assigned as 15 and 10 in all of our experiments. To optimize this energy, we use the similar coordinate-descent with the method in identity fitting: in each iteration, we first optimize rigid rotation and translation while fixing expression coefficients; and vice versa. During optimizing the blendshape coefficients, different from fitting identity, we need to constrain the range of each coefficient β_i. We use the gradient projection algorithm based on the BFGS solver [29] to restrict the range of expression coefficients in $[0, 1]$. With the solved $\mathbf{R}^t$, $\mathbf{t}^t$ and $\mathbf{b}^t$, we can finally generate the tracked 3D shape (Fig.3(d)).

5. EXPERIMENTS

5.1. LVCSR based phoneme recognition and alignment

Our evaluation set is a series of videos that contain clean speech and clear frontal faces. Our acoustic model is trained with the fisher corpus [30], which has about 1761 hours of data, Kaldi [31] and its Aspire receipt is used for acoustic model training. The model and speech recognition structure is described in Section 3.1. Since BLSTM setting requires more future context and suffers from the inability of online decoding, and the performance difference as compared to TDNN setting is minimal, we therefore use the TDNN setting in all our experiments. A tri-gram language model is trained with 1.8B words from multiple text corpus we collected. The vocabulary size is 123K. Compared to a language model trained with fisher

	Word Error Rate	Phoneme Error Rate
TIMIT	-	8.7%
Evaluation Set	4.9%	2.5%

Table 1. Result of a word error rate and phoneme error rate on two evaluation dataset.

corpus [31], our language model improves WER on our evaluation set by 5.6% absolutely.

We use several different datasets to benchmark the phoneme recognition performance. The estimate of Phoneme Error Rate (PER) for TIMIT differs from traditional PER reported in most of the literatures [32]. Traditionally, phoneme sequence is generated through a phoneme recognizer acoustic model and normally includes a bi-gram phoneme language model trained on this specific dataset. The metrics are calculated against the ground truth phoneme label. We first perform word level recognition and then use a dictionary to expand the word sequence to the phoneme sequence and calculate its corresponding PER. As shown in Tab. 1, our phoneme error rate is only 8.7%. While on a similar 39 phoneme scale, the reported PER is between 18-30% with different settings. We also tested on the video clips we random collected, and we showed a word error rate of 4.9% and phoneme error rate of 2.5%.

5.2. Objective: MSE compares to keyed shape by animator

We ask a professional animator to key the facial expression coefficients for one video. We then compare the MSE between the keyed 3D shape and our generated shape using different methods respectively. As seen from Tab.2, our joint audio-video driven method can achieve the lowest error.

Method	Audio	Video	Audio-video
Error (mm)	7.53	5.16	4.83

Table 2. The error of tracked 3D shapes with animator keyed shape.

5.3. Subjective user study

We conducted subjective tests and the results presented below are based on 18 unique inputs. First, we aim to compare the audio-driven lip motion synthesis quality against different phoneme recognition techniques. As similar to [15], we use the TIMIT corpus. But we train a neural network acoustic model as in [2] instead of HMM/GMM model.(referred as Phone_NN.) The training data set consists of 3696 utterances from 462 speakers. We use the 39 phoneme set up. The acoustic features are 13 MFCCs that are extracted with a 10ms shift. A context of 17 Frames of feature, shaped by mean variance normalization, is used by DNN as the input. The input dimension is 221. Three fully connected layers with each layer containing 1024 hidden nodes are used, and the output layer contains 39 phoneme classes. TNet [33] is used to conduct the training. This model is capable to support real-time decoding on device. We evaluated this on one of our video clips and result summarized in the Tab.2, where 88.9% of test takers prefer the LVCSR generated animation.

	Phoneme NN	No difference	LVCSR
user preferences	0%	11.1%	88.9%

Table 3. Subjective test on user preferences comparing Phoneme_NN and LVCSR

In the second experiment, we **compare audio driving vs. video driving vs. joint audio-video driving**. As showed in Tab.4, our proposed joint audio-video show superior preferences comparing to audio only or video only. We also show some example frames at Fig.4. Our method can achieve the best results in all frames.

	Audio	Video	Audio-video
Least preferred	63.9%	31.9%	4.2%
Middle	19.4%	48.6%	31.9%
Most preferred	16.7%	20.6%	63.9%

Table 4. Subjective test on user preferences comparing audio driving vs. video driving vs. joint audio-video driving.

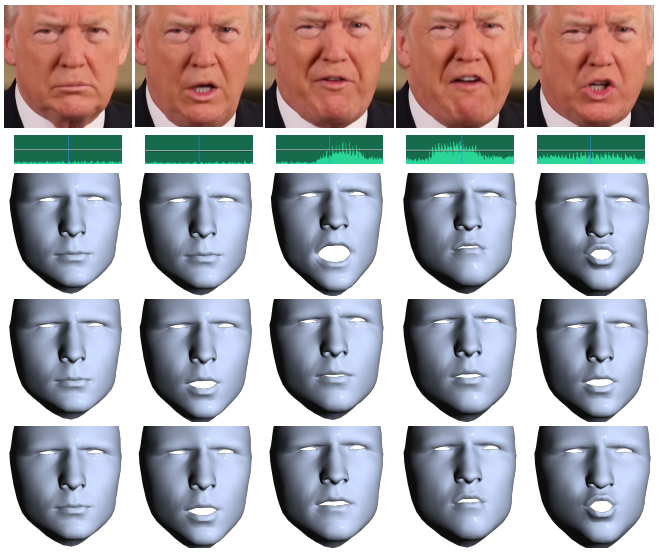


Fig. 4. Comparisons of results by different methods. Top two rows: input video and audio. Bottom three rows, from top to bottom: results by audio-driven, video-driven and our joint audio-video driven. From left to right: frame number 1, 15, 29, 184, 236.

In the third experiment, we **compare ours results vs. professional animator keyed results**, we evaluated this on one selected video clips and the result is summarized in the Tab.5, the animator keyed is slightly preferred here, but the difference is very small.

	animator keyed	No difference	Ours
user preferences	33.3%	38.9%	27.8%

Table 5. Subjective test on user preferences comparing Artist Keyed between Our proposed method

In the final experiment, we **compare ours results vs. [2]**. We evaluate one of [2]'s video clips, and results are summarized in the Tab.6. As showed in the table, the majority of the test takers prefer our results. But [2]'s result is real-time processed, while currently our result is not.

	[2]	No difference	Ours
user preferences	16.7%	0%	83.3%

Table 6. Subjective test on user preferences comparing [2]'s result and our proposed method

6. CONCLUSION

We present a joint audio-video driven facial animation system. Our system can robustly estimate the phoneme with a state-of-the-art LVCSR. With the guides of 3D face scans, we build a mapping from phonemes to 3D blendshapes. Then our system combines both the audio and video information to generate facial animation. Experiments show that our proposed approach obtains strong results and its quality surpasses that of some competitive systems [2]. In the future, we plan to apply this technology to improve facial animation quality and efficiency for animated content production.

7. REFERENCES

[1] N. Ersotelos and F. Dong, "Building highly realistic facial modeling and animation: a survey," in *Visual Computer*, 2008, vol. 28, pp. 13–30.

[2] Yilong Liu, Feng Xu, Jinxiang Chai, Xin Tong, Lijuan Wang, and Qiang Huo, "Video-audio driven real-time facial animation," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 6, pp. 182, 2015.

[3] Thibaut Weise, Sofien Bouaziz, Hao Li, and Mark Pauly, "Realtime performance-based facial animation," in *ACM SIGGRAPH 2011 Papers*, New York, NY, USA, 2011, SIGGRAPH '11, pp. 77:1–77:10, ACM.

[4] Sofien Bouaziz, Yangang Wang, and Mark Pauly, "Online modeling for realtime facial animation," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 40:1–40:10, July 2013.

[5] Hao Li, Jihun Yu, Yuting Ye, and Chris Bregler, "Realtime facial animation with on-the-fly correctives," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 42:1–42:10, July 2013.

[6] Chen Cao, Yanlin Weng, Stephen Lin, and Kun Zhou, "3d shape regression for real-time facial animation," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 41:1–41:10, July 2013.

[7] Chen Cao, Qiming Hou, and Kun Zhou, "Displaced dynamic expression regression for real-time facial tracking and animation," *ACM Trans. Graph.*, vol. 33, no. 4, pp. 43:1–43:10, July 2014.

[8] John Lewis, "Automated lip-sync: Background and techniques," *Computer Animation and Virtual Worlds*, vol. 2, no. 4, pp. 118–122, 1991.

[9] Changwei Luo, Jun Yu, and Zengfu Wang, "Synthesizing real-time speech-driven facial animation," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 4568–4572.

[10] G. Takacs, "Direct, modular and hybrid audio to visual speech conversion methods - a comparative study," in *Proc. Interspeech*, 2009.

[11] W. Han, L. Wang, F. Soong, and B. Yuan, "improved minimum converted trajectory error traing for real-time speech-to-lips conversion," in *ICASSP 2012 Proceedings*, 2012.

[12] Tsuhan Chen and Ram R. Rao, "Audio-visual interaction in multimedia communication," in *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*. IEEE, 1997, vol. 1, pp. 179–182.

[13] L. Xie and Z. Q. Liu, "A coupled hmm approach to video-realistic speech animation," in *Pattern Recognition 40*, 2007, vol. 8, pp. 2325–2340.

[14] S. Taylor, T. Kim, Y. Yue, M. Mahler, J. Krahe, A. G. Rodriguez, J. Hodgins, and I. Matthews, "A deep learning approach for generalized speech animation," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, pp. 93, 2017.

[15] C. Bregler, M. Covell, and Slaney M., "Video rewrite: driving visual speech with audio," in *Proceedings of ACMSIGGRAPH*, 1997.

[16] Y. Cao, W. C. Tien, P. Faloutsos, and F. Pighin, "Expressive speech-driven facial animation.," in *ACM Trans*, 2005, vol. Graph.24, 4(Oct), pp. 1283–1302.

[17] Z. Zhang, L. Wang, G. Li, F. Seide, and F. K. Soong, "A new language independent, photo-realistic talking head driven by voice only," in *INTERSPEECH*, 2013, pp. 2743–2747.

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

[19] Alex Waibel, Toshiyuki Hanazawa, Geoffrey Hinton, Kiyohiro Shikano, and Kevin J Lang, "Phoneme recognition using time-delay neural networks," *IEEE transactions on acoustics, speech, and signal processing*, vol. 37, no. 3, pp. 328–339, 1989.

[20] Daniel Garcia-Romero and Carol Y Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems.," in *Interspeech*, 2011, vol. 2011, pp. 249–252.

[21] Cyril Allauzen, Michael Riley, Johan Schalkwyk, Wojciech Skut, and Mehryar Mohri, "Openfst: A general and efficient weighted finite-state transducer library," *Implementation and Application of Automata*, pp. 11–23, 2007.

[22] Robert L Weide, "The cmu pronouncing dictionary," *http://www. speech. cs. cmu. edu/cgibin/cmudict*, 1998.

[23] "Bellus3d: High-quality 3d face scanning," *http://www.bellus3d.com/*.

[24] Chen Cao, Yanlin Weng, Shun Zhou, Yiying Tong, and Kun Zhou, "Facewarehouse: A 3d facial expression database for visual computing," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 3, pp. 413–425, Mar. 2014.

[25] Paul Ekman and Erika L Rosenberg, *What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS)*, Oxford University Press, USA, 1997.

[26] Vahid Kazemi and Josephine Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *CVPR*, 2014.

[27] Volker Blanz and Thomas Vetter, "A morphable model for the synthesis of 3d faces," in *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, New York, NY, USA, 1999, SIGGRAPH '99, pp. 187–194.

[28] Robert W. Sumner and Jovan Popović, "Deformation transfer for triangle meshes," in *ACM SIGGRAPH 2004 Papers*, New York, NY, USA, 2004, SIGGRAPH '04, pp. 399–405, ACM.

[29] Richard H. Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu, "A limited memory algorithm for bound constrained optimization," *SIAM J. Sci. Comput.*, vol. 16, no. 5, pp. 1190–1208, Sept. 1995.

[30] Christopher Cieri, David Miller, and Kevin Walker, "The fisher corpus: a resource for the next generations of speech-to-text.," in *LREC*, 2004, vol. 4, pp. 69–71.

[31] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, et al., "The kaldi speech recognition toolkit," in *IEEE workshop on ASRU*, 2011.

[32] Tuo Zhao, Yunxin Zhao, and Xin Chen, "Time-frequency kernel-based cnn for speech recognition," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[33] K. Vesely, L. Burget, and F. Grezl, "Parallel training of neural networks for speech recognition," *Proc. of Interspeech*, 2010.