High Quality Lips Animation with Speech and Captured Facial Action Unit as A/V Input

Lijuan Wang and Frank K. Soong
Microsoft Research Asia, Beijing, China
E-mail: {lijuanw, frankkps}@microsoft.com Tel: +86-10-59174988

***Abstract*—Rendering realistic lips movements in avatar with camera captured human's facial features is desirable in many applications, e.g. telepresence, video gaming, social networking, etc. We have proposed to use Gaussian Mixture Model (GMM) to generate lips trajectory and successfully tested in speech-to-lips conversion experiments, where only audio signal (speech) is used as input. In this paper real-time user's facial features called the Action Units (AUs) well tracked by Microsoft Kinect SDK with a consumer-grade RGB camera, are combined with speech to form joint A/V input for lips animation. We test the lips animation performance and show that the new combined A/V input can improve the conversion error rate by 22% in a speaker dependent test, compared with a baseline system.**

I. INTRODUCTION

Lips animation retargeting can transfer lips movement of a person captured by a camera to an avatar in real-time, which offers a wide range of useful applications and generating such lips animation is useful for human computer interaction in video games or other augmented reality scenarios. As audio and visual information co-occur in human communications, animated avatar can also benefit human-human interactions, e.g. internet videophones in low bandwidth. Other applications include an avatar based video conference where actual captured video will not be shown due to some privacy concern.

Previous work e.g. [1][2], used facial feature point provided by a video tracker or a motion capture system to retarget user's facial animation to a cartoon-like avatar. However, as avatar becomes more human-like and photo-realistic, retargeting mouth area animation becomes quite a technical challenge. This is because mouth is a combination of delicate tissues of different types, including: lips, tongue, and teeth. Moreover, lips movement is quicker than any other facial muscles when speaking, and sometime with occlusions of tongue and teeth. Using tracked feature point (usually lips contours) cannot provide enough information for animation of lips, tongue, and teeth.

A different approach, called speech-to-lips conversion, uses speech as the source signal and converts it directly into visual lips movement,. In speech-to-lips conversion, we establish a mapping between acoustic speech space and visual mouth space. In other words, given the acoustic parameters of speech, one needs to estimate the corresponding mouth parameters and/or vice versa. The conversion is to find the best mapping, for given dual training sets.

Numerous attempts to model the relationship between audio (speech) and visual (usually lips, sometimes also upper face) signals and many are generative probabilistic model based, where the underlying probability distributions of audio-visual data is estimated. Typical model assumptions are Gaussian Mixture model (GMM), Hidden Markov Model (HMM) [3], Dynamical Bayesian Network (DBN) [4] and Switching Linear Dynamical System (SLDS) [5], in increasing model complexity. In our previous work [8], we proposed Minimum Converted Trajectory Error training which, unlike maximum likelihood criteria, minimizes the converted trajectory error over training data so as to improve the quality conversion.

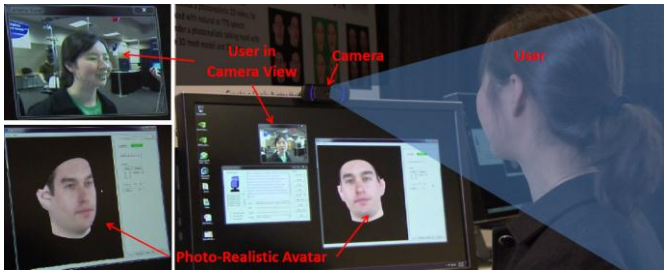


Fig. 1: Retargeting user's facial animation to a photo-realistic avatar.

The drawback of speech-to-lips conversion is that it is not robust against environmental noises, difference across microphone channels, and speaker change, etc. All the above acoustic uncertainties have no effect on tracked facial features. On the other hand, illumination variation, which is highly sensitive in facial feature tracking, will not affect speech-to-lips conversion. Therefore, it is desirable if we can combine speech and captured (tracked) facial features as joint A/V input and use them for lips animation retargeting. In this paper, we propose to use facial action units estimated from face video combined with audio feature to improve conversion performance. Experimented results show that facial animation unit can reduce the conversion error by 22% in a speaker dependent task.

The rest of the paper is organized as follows. In Section II we propose the whole system and application scenarios. In Section III we introduce facial action units and how to estimate AUs from face video. In Section IV we introduce Maximum Likelihood (ML) and Minimum Converted Trajectory Error (MCET) criteria for training. Section V presents the experimental results. Section VI is the conclusion.

II. SYSTEM OVERVIEW

The conversion system has two, model training and conversion, stages, as shown in Fig.2. First of all, a parallel audio-video database is recorded and serves as training data. Acoustic feature MFCC is extracted from the audio stream. Video face tracking is carried out on the video stream, after that facial action units can be estimated for every image frame describing the simplified animation pattern of lips motion. PCA Eigen lips are also extracted from the video image sequence, representing the detailed appearance features. The three features: MFCCs, AUs, and PCAs are augmented into a super feature vector sequence. With these features, a statistical GMM is automatically trained to characterize the joint feature space.

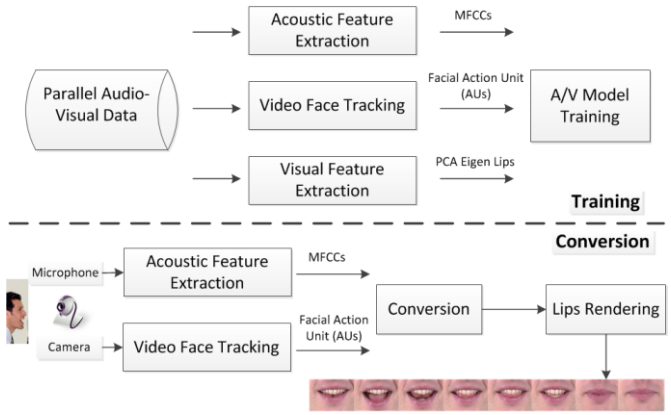


Fig. 2: Lips rendering with speech and AUs as A/V input.

In the conversion stage, with a normal RGB camera setup, user's face video and audio speech can be captured synchronously. After applying acoustic feature extraction and video face tracking, MFCCs and AUs can be obtained and combined as source input to generate the most likely PCAs parameter trajectories representing the model estimated lips animation. The converted PCA trajectories are used in rendering photo-realistic lips movement for an avatar.

III. Face Action Unit (AU)

A. *Facial Action Coding System*

Rapid facial movements formed when facial muscles pull the skin, causing a temporary distortion of the shape of the facial features and of the appearance of folds, furrows, and bulges of skin. The common terminology for describing rapid facial movement signal refers to either culturally dependent linguistic terms indicating a specific appearance change of a particular facial feature (e.g., smile, smirk, frown, sneer) or the linguistic universals describing the activity of specific facial muscles that caused the observed facial appearance changes. There are several methods for linguistically universal recognition of facial changes based on the facial muscular activity. From those, the facial action coding system (FACS) proposed by Ekman et al. [12] is the best known and most commonly used system. It is a system designed for human observers to describe changes in the facial expression in terms of visually observable activation of facial muscles. The changes in the facial expression are described with FACS in terms of 44 different Action Units (AUs), each of which is anatomically related to the contraction of either a specific facial muscle or a set of facial muscles. Examples of different AUs are shown in Fig.3.

B. *Facial Action Units*

We adopt the face tracking library Microsoft released as part of Kinect for Windows SDK. The SDK can be used with a normal RGB camera or Kinect sensor. The real-time face tracking engine can output the 2D feature points and 3D head pose about a tracked user, as shown in Fig.4. The Face Tracking Library's results are also expressed in terms of weights of six Action Units (AUs) and eleven Shape Units (SUs), which are a subset of what are defined in the Candide3 model [13]. The Shape Units estimate the particular shape of the user's head: the neutral position of their mouth, brows, eyes, etc. The shape units, i.e. the neutral shape of a specific tracked user, are estimated from the first few frames and then keep unchanged in the rest frames. The action units are changes from the neutral shape which can drive morph targets in avatar models to produce corresponding animations. Among all facial action units, four AUs relating to lips motion (shown in Fig.5) are chosen as source visual input in conversion. Although AUs are invariant to pose and face shape difference, the animation pattern can still be quite different between two speakers. The normalized histograms of four AUs over same 200 utterances of two speakers are plotted in Fig.6. Different distri-

AU1	AU2	AU4	AU5	AU6
Inner brow raiser	Outer brow raiser	Brow lower-er	Upper lid raiser	Cheek raiser
AU7	AU9	AU10	AU12	AU15
Lid tightener	Nose wrin-kler	Upper lip raiser	Lip Corner Puller	Lip corner depressor
AU17	AU18	AU20	AU22	AU28
Chin raiser	Lip puckerer	Lip stretcher	Lip funneler	Jaw drop

Fig. 3: Some examples of Facial Action Units.

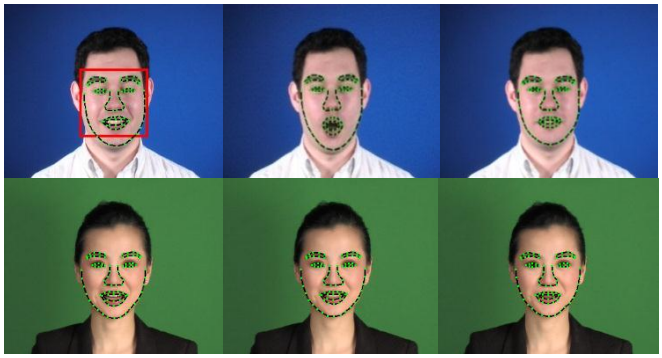


Fig. 4: Face tracking results on two speakers' face video.

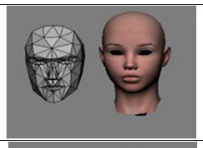
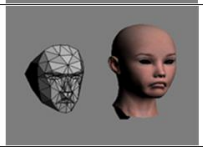
Neutral Face (all AUs 0)		
AU0 – Upper Lip Raiser (In Candid3 this is AU10)		0=neutral, covering teeth; 1=showing teeth fully
AU1 – Jaw Lowerer (In Candid3 this is AU26/27)		0=closed; 1=fully open
AU2 – Lip Stretcher (In Candid3 this is AU20)		0=neutral; 1=fully stretched (joker's smile); -0.5=rounded (pout)
AU4 – Lip Corner Depressor (In Candid3 this is AU13/15)		0=neutral; -1=very happy smile; +1=very sad frown

Fig. 5: Four AUs (Facial Action Units) for describing lips motion.

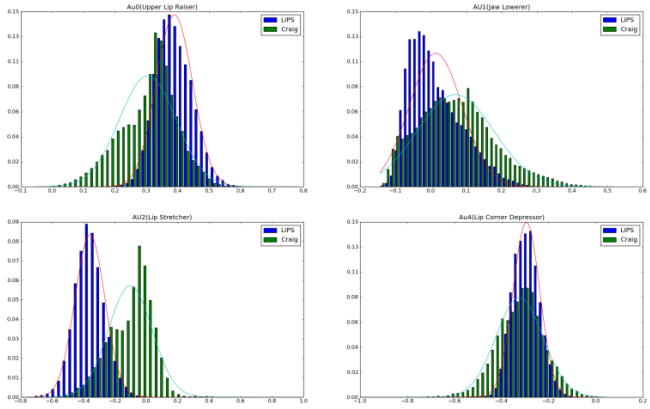


Fig. 6: Distribution difference of AUs between two speakers.

butions suggest speakers may choose different lips action units in articulating same sentences.

IV. GMM-Based X-to-Y Conversion

We generalize the problem as X-to-Y conversion. The best estimate of Y for given X is derived directly from the joint statistics of X and Y. With the joint probability distribution of {X,Y} vectors, estimated as a Gaussian mixture model, we can derive the optimal estimate of Y given X analytically. The X-to-Y conversion consists of two, training and conversion, stages. In training, X and Y features are extracted as parallel training vectors in two corresponding feature spaces. A GMM is automatically trained to characterize the joint distribution. In conversion, for any given sequences in X space can be mapped to Y space.

A. GMM-based conversion under ML criterion

We denote the X and Y sequence, and their time derivative as,

$$x = [x_1^\intercal, \cdots, x_T^\intercal]^\intercal, \qquad y = [y_1^\intercal, \cdots, y_T^\intercal]^\intercal,$$
$$\Delta x_i = \frac{1}{2}(x_{i+1} - x_{i-1}), \qquad \Delta y_i = \frac{1}{2}(y_{i+1} - y_{i-1}),$$

$$X_i = [x_i^\intercal, \Delta x_i^\intercal], \qquad Y_i = [y_i^\intercal, \Delta y_i^\intercal],$$
$$X = [X_1^\intercal, \cdots, X_T^\intercal]^\intercal, \qquad Y = [Y_1^\intercal, \cdots, Y_T^\intercal]^\intercal,$$

where x and y can be acoustic, AU, and visual feature sequence of an utterance. We further augment the feature vector with its time derivative (or dynamic features [9]) Δx_i and Δy_i, so that,

$$X = Wx, Y = Wy$$

In the GMM based approach, every X_i and Y_i are assumed to be independently drawn from a mixture of Gaussian distributions,

$$P(X_i, Y_i|\lambda) = \sum_{m=1}^{M} w_m \mathcal{N}(X_i, Y_i; \mu_m, \sum_m) \tag{1}$$

where m is the index of Gaussian mixture component, μ_m and $\sum_m$ denote the mean and covariance of the m^{th} Gaussian.

With a trained GMM, X-to-Y conversion of a sequence is formulated as,

$$P(\mathrm{Y}|X, \lambda) = \prod_{t=1}^{T} P(Y_t|X_t, \lambda)$$
$$= \prod_{t=1}^{T} \sum_{m_t=1}^{M} P(m_t|X_t, \lambda) P(Y_t|X_t, m_t, \lambda) \tag{2}$$

$$\hat{y} = argmax\, P(Y|X) \tag{3}$$

In practice, we make several approximations to reduce the complexity in solving Eq. 3. First, the summation in Eq. 3 is approximated by the Maximum A Posterior (MAP) mixture component,

$$P(\mathrm{Y}|X, \lambda) \approx \prod_{t=1}^{T} P(\hat{m}_t|X_t, \lambda) P(Y_t|X_t, \hat{m}_t, \lambda) \tag{4}$$
$$\hat{m}_t = \arg max P(m|X_t, \lambda)$$

With this approximation, Eq. 4 can be solved in a closed form,

$$\hat{y} = \left(\mathrm{W}^\intercal D_{\hat{m}}^{(Y)^{-1}} W\right)^{-1} W^\intercal D_{\hat{m}}^{(Y)^{-1}} E_{\hat{m}}^{(Y)} \tag{5}$$

where

$$E_{\hat{m}}^{(Y)} = \left[E_{\hat{m}_1}^{(Y)}, \cdots, E_{\hat{m}_T}^{(Y)}\right] \tag{6}$$
$$D_{\hat{m}}^{(Y)^{-1}} = diag\left[D_{\hat{m}_1}^{(Y)^{-1}}, \cdots, D_{\hat{m}_T}^{(Y)^{-1}}\right] \tag{7}$$

and

$$E_{\hat{m}_t}^{(Y)} = \mu_{\hat{m}_t}^{(Y)} + \sum_{\hat{m}_t}^{(YX)} \sum_{\hat{m}_t}^{(XX)^{-1}} \left(X_t - \mu_{\hat{m}_t}^{(X)}\right) \tag{8}$$
$$D_{\hat{m}}^{(Y)} = \sum_{\hat{m}_t}^{(YY)} - \sum_{\hat{m}_t}^{(YX)} \sum_{\hat{m}_t}^{(XX)^{-1}} \sum_{\hat{m}_t}^{(XY)} \tag{9}$$

Second, to have a robust estimation of covariance matrix $\sum$, we assume the off-diagonal terms in $\sum_{\mathrm{m}}^{(\mathrm{XY})}$ and $\sum_{\mathrm{m}}^{(\mathrm{YX})}$ to be all null, and $\sum_{\mathrm{m}}^{(\mathrm{XX})}$ and $\sum_{\mathrm{m}}^{(\mathrm{YY})}$ to be diagonal. In other words, correlations between different dimensions in the joint X-Y feature space are ignored. Eventually, Eq. 8 and Eq. 9 are simplified to,

$$E_{\hat{m}_t}^{(Y)} = \mu_{\hat{m}_t}^{(Y)}, D_{\hat{m}}^{(Y)} = \sum_{\hat{m}_t}^{(YY)}. \tag{10}$$

B. MCTE-based conversion

MCTE aims to minimize the error of the converted trajectory, i.e., Euclidean distance between the converted trajectory and the ground truth over all training data.

$$D(y, \hat{y}) = \|y - \hat{y}\|_2^2 = \sum_{t=1}^{T} \|y_t - \hat{y}_t\|_2^2 \tag{11}$$

$$L(\lambda) = \frac{1}{N} \sum_{i=1}^{N} D\left(y^i, \hat{y}^i\right) \tag{12}$$

Note that in GMM conversion with MAP approximation Eq. 4, the conversion is actually accomplished in two steps. First, a sequence of Gaussian mixtures is estimated from observation X by MAP: $\hat{m} = arg\, maxP(m|X, \lambda)$. Then, visual trajectory, $\hat{y}$, is generated from the mixture component sequence $\hat{\mathrm{m}}$ by maximizing $P(Wy|\hat{m}, \lambda)$, leading to the closed form solution in Eq. 5. Thus, the MCTE loss function Eq. 12 becomes a function of $\lambda^{(\mathrm{Y})}$ for given mixture sequence $\hat{\mathrm{m}}$. We minimize it by probabilistic descend (PD) algorithm.

The probabilistic descend (PD) algorithm update model parameters at each training utterance. For the n^{th} utterance,

$$\lambda_{n+1}^{(Y)} = \lambda_n^{(Y)} - \varepsilon_n \frac{\partial \mathrm{D}(\mathrm{y}^n, \hat{\mathrm{y}}^n)}{\partial \lambda^{(Y)}}\Big|_{\lambda^{(Y)}=\lambda_n^{(Y)}}$$
$$= \lambda_n^{(Y)} - 2\varepsilon_n (\mathrm{y}^n - \hat{\mathrm{y}}^n)^\intercal \frac{\partial \hat{\mathrm{y}}^n}{\partial \lambda^{(Y)}} \tag{13}$$

By Eq. 5,

$$\frac{\partial \hat{\mathrm{y}}^n}{\partial \mathrm{E}_{\hat{m}_t,d}^{(Y)}} = \left(\mathrm{W}^\intercal D_{\hat{m}}^{(Y)^{-1}} W\right)^{-1} W^\intercal D_{\hat{m}}^{(Y)^{-1}} Z_E \tag{14}$$

where $E_{\hat{m}_t,d}^{(Y)}$ is the d^{th} dimension of the mean vector of the t^{th} mixture in the MAP mixture sequence, $Z_E = [0, \ldots, 0, 1_{1\times D_Y+d}, 0, 0, \ldots, 0]^\intercal$ and D_Y is the dimension of Y.

For convenience we denote $v_{t,d} = 1/\sigma_{t,d}^2$, $Z_v = Z_E Z_E^\intercal$. $\sigma_{t,d}^2 = D_{\hat{m}_t,d}^{(Y)}$ is the variance corresponding of $E_{\hat{m}_t,d}^{(Y)}$. The updating rule for covariance is,

$$\frac{\partial \hat{y}^n}{\partial v_{t,d}} = \left(W^T D_{\hat{m}}^{(Y)^{-1}} W\right)^{-1} W^T Z_v \left(E_{\hat{m}}^{(Y)} - W\hat{y}^n\right). \quad (15)$$

V. Experimental Results

A. Experimental Setup

The database used in LIPS 2008 Visual Speech Synthesis Challenge [11] was used in our experiments. It consists of 278 video clips, each is an English sentence recorded neutrally by a female, native speaker of British English. Videos were recorded in 50 FPS. The acoustic features are Mel-frequency Cepstral Coefficient (MFCC) extracted from speech in a 20ms window, shifted every 5ms. From the first few frames, the neutral shape of the speaker is estimated and the shape units are fixed for all the frames in the database. The AUs are estimated for every frame and the four AUs relevant to lips motion are used as complementary input for both training and testing. For visual feature we perform Principal Component Analysis (PCA) on the automatically detected and aligned mouth image, keep the first 20 principal components. Fig. 7 shows the four AUs and the first PCA parameters of a training sentence video, from which we can see the correlation between AUs and PCA parameters. Both AUs and PCA visual feature vectors are interpolated to the same frame rate as audio speech MFCCs.

The objective evaluations are performed with two metrics. First, we use all data for both training and conversion in evaluating the "training" performance. For the "testing" performance, we perform a leave-20-out cross validation, and the results of all folds are averaged as the "testing" performance. The conversion performances are evaluated using Mean Square Error (MSE), defined as follows,

$$MSE = \frac{1}{T}\sum_{t=1}^{T} \|\hat{y}_t - y_t\| \quad (16)$$

B. Speaker dependent experimental results

While two source signals, speech and AUs, were used in conversion, three different conversion modes, speech, AU, and combined, are investigated. In each mode, training and testing are the same, but different signal are used as input in the conversion. In speech mode, the conversion is between 39-dimension speech features (MFCC+ΔMFCC +ΔΔMFCC) and visual PCA parameters. In AU mode, the conversion source signal is 12-dimension (AUs+ΔAUs+ΔΔAUs) and AU owns all the stream weights in ML-based GMM training. In combined mode, the speech MFCC feature is augmented by the 12-dimensional AU feature. In ML-based GMM training, the AU and MFCC are equally weights. Table 1 shows the mean square errors between the converted PCA trajectory and the ground truth trajectory. It shows that by combining facial action units with speech can further reduce the conversion errors by 22%.

Table 1: MSE of the conversion error in three modes: speech, AUs, and their combination.

Mode	ML closed	MCTE closed	ML open	MCTE open
Speech	8.93E5	5.21E5	9.36E5	7.62E5
AUs	6.03E5	4.47E5	7.23E5	6.45E5
AUs + Speech	6.95E5	4.47E5	7.08E5	**5.96E5**

VI. Conclusions

We propose to use speech and facial action unit captured from face video as A/V input to synthesize high quality, realistic lips movement in a photo-realistic avatar. Experimental results show that the new approach can reduce the conversion error by 22% on a speaker dependent task.

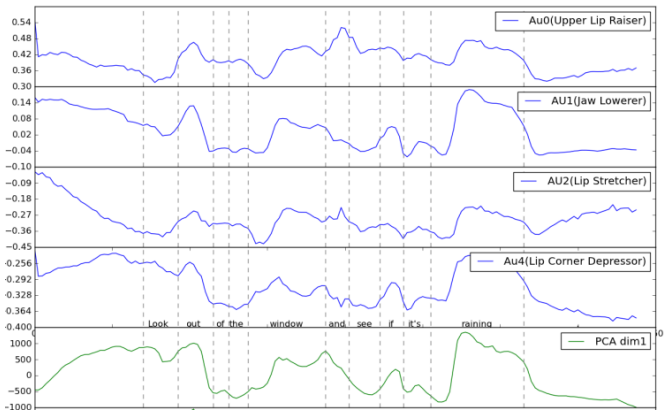


Fig. 7: Trajectories of AUs and the first PCA for an utterance.

VII. References

[1] L. Dutreve, A. Meyer, S. Bouakaz, "Feature Points based Facial Animation Retargeting," In Proc. of the 2008 ACM symposium on Virtual reality software and technology (2008), pp. 197-200.

[2] R. Seguier, G. Breton, N. Stoiber, "Facial Animation Retargeting and Control Based on a Human Appearance Space," Computer Animation and Virtual Worlds, Vol. 21, No.1, pp.39-54.

[3] S. Fu, R. Gutierrez-Osuna, A. Esposito, P. K. Kakumanu, and O. N. Garcia, "Audio/visual mapping with cross-modal hidden Markov models," IEEE Trans. On Multimedia, vol. 7, no. 2, pp. 243-252, 2005.

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

[5] G. Englebienne, T. F. Cootes, and M. Rattray, "A probabilistic model for generating realistic lip movements from speech," in NIPS, 2007.

[6] K. Grant and S. Greenberg, "Speech intelligibility derived from asynchronous processing of auditory-visual information," in AVSP 2001-International Conference on Auditory-Visual Speech Processing, 2001, pp. 132-137.

[7] T. Toda, A. Black, and K. Tokuda, "Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory," IEEE Trans. On Speech and Audio Processing, vol. 15, no. 8, pp. 2222-2235, 2007.

[8] X. Zhuang, L. Wang, F. K. Soong, and M. Hasegawa-Johnson, "A minimum converted trajectory error (MCTE) approach to high quality speech-to-lips conversion," in Interspeech, 2012, pp. 1736-1739.

[9] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," in ICASSP, 2000, pp. 1315-1318.

[10] Y.-J. Wu and R.-H. Wang, "Minimum generation error training for HMM-based speech synthesis," in ICASSP, 2006, pp. 89-92.

[11] B.-J. Theobald, S. Fagel, G. Bailly, and F. Elisei, "LIPS2008: visual speech synthesis challenge," in Interspeech, 2008, pp. 2310-2313.

[12] P. Ekman and W. Friesen, "Facial Action Coding System: A Technique for the Measurement of Facial Movement," Consulting Psychologists Press, Palo Alto, 1978.

[13] Candide3 model: http://www.icg.isy.liu.se/candide/