SYNTHESIZING REAL-TIME SPEECH-DRIVEN FACIAL ANIMATION

Changwei Luo[1], *Jun Yu*[1], *Zengfu Wang*[12]

[1]Department of Automation, University of Science and Technology of China, P.R.C
[2]Institute of intelligent machines, Chinese Academy of Sciences, P.R.C
luocw@mail.ustc.edu.cn, {harryjun, zfwang}@ustc.edu.cn

ABSTRACT

We present a real-time speech-driven facial animation system. In this system, Gaussian Mixture Models (GMM) are employed to perform the audio-to-visual conversion. The conventional GMM-based method performs the conversion frame by frame using minimum mean square error (MMSE) estimation. The method is reasonably effective. However, discontinuities often appear in the sequences of estimated visual features. To solve this problem, we incorporate previous visual features into the conversion so that the conversion procedure is performed in the manner of a Markov chain. After audio-to-visual conversion, the estimated visual features are transformed to blendshape weights to synthesize facial animation. Experiments show that our system can accurately convert audio features into visual features. The conversion accuracy is comparable to a current state-of-the-art trajectory-based approach. Moreover, our system runs in real time and outputs high quality lip-sync animations.

Index Terms: audio-to-visual conversion, GMM, blendshape, facial animation

1. INTRODUCTION

Facial animation is a hot research topic in both academia and industry, its applications include movie industry, computer games and human-computer interaction [1]. In human-computer interaction applications, a lip-sync talking head can attract the attention of a user, and make human-machine interaction more effective. It is reported that the trust and attention of humans towards machines are able to increase by 30 percent if humans are communicating with talking heads instead of text-only [2].

A number of researchers have described techniques for synthesizing realistic lip-sync animations [3, 4]. Ezzat et al. [5] synthesize speech animation by using a recorded video database. The coarticulation effects are represented by the magnitude of diagonal covariance matrices of phoneme clusters. Wang et al. [6] propose a system which renders a photo-real video of articulators in sync with the given speech by searching for the most plausible real image sample sequence. These methods achieve high realism in synthesized videos. However, it is challenging to change head pose freely or to render different lighting conditions. Deng et al. [7] first construct explicit speech coarticulation models from real human motion data, then new speech animations are synthesized by blending 13 key viseme shapes.

According to the input of the animation systems, talking heads can be text-driven or speech-driven. Although most text-driven talking heads employ advanced speech synthesizers [8], they still lack natural speech prosody and emotions. Therefore we focus on synthesizing facial animation from real human speech.

Audio-to-visual conversion is the core of speech-driven facial animation. Various approaches have been proposed to model the relationship between audio and visual features. Hidden Markov models are widely used in audio-to-visual conversion [8, 9], an advantage of the these approaches is that context information can be easily represented by state-transition probabilities. However, they usually require a phoneme sequence that is provided by an automatic speech recognizer. The synthesis performance heavily depends on the Viterbi search. The Viterbi sequence may represent only a small fraction of the total probability mass, and many other slightly different state sequences potentially have nearly equal likelihoods [10].

In contrast to the phoneme-based conversion, direct conversion without using phonetic information, is also effective. The work presented in [11] shows that automatic speech recognizer based conversion is inferior to direct conversion in their experiments when a neural network is used. Besides neural networks, GMM has also been used for direct conversion [12]. Toda et al. [13, 14] have tested two different GMM-based methods: the conventional method [15] and the trajectory-based method. The former runs in real time with relatively lower performance, the latter has better performance but it comes with a latency time no less than the length of one utterance.

In this paper, we also use the GMM-based method for audio-to-visual conversion. The conventional GMM-based method works in real time. However, the performance of the conversion is insufficient, discontinuities often appear in the sequences of the estimated target vectors. To solve this problem, we incorporate previous visual features into the conversion. Based on the proposed method, we develop a real-time speech-driven facial animation system.

2. REAL-TIME GMM-BASED CONVERSION

2.1. The conventional method

The conventional method converts source features into target features frame by frame using MMSE estimation. Let x_t and y_t be the source and target feature vectors at frame t, respectively. The joint probability density of the source and target vectors can be modeled by a GMM as follows:

$$P(z_t|\lambda^{(z)}) = \sum_{m=1}^{M} w_m \cdot N(z_t\,;\,\mu_m^{(z)}, \Sigma_m^{(z)}), \qquad (1)$$

where $z_t = [x_t\,;\,y_t]$ is a joint vector, the total number of mixture components is M. A parameter set of the GMM is $\lambda^{(z)}$, the mean vector $\mu_m^{(z)}$ and the covariance matrix $\Sigma_m^{(z)}$ of the m^{th} mixture com-

This work is supported by the National Natural Science Foundation of China (No. 61303150), the Fundamental Research Funds for the Central Universities (No. WK2100100020), and the STP of Anhui (No.11010202192).

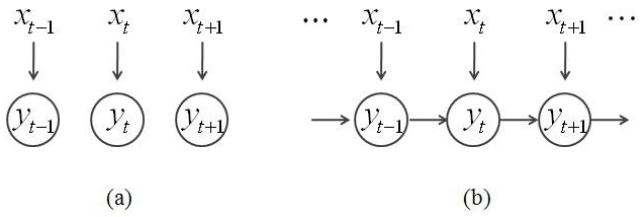

Fig. 1: Audio-to-visual Conversion. (a) The current visual feature only depends on the audio feature, (b) The current visual feature depends on not only the audio feature but also its previous visual feature.

ponent are written as:

$$\mu_m^{(z)} = \begin{bmatrix} \mu_m^{(x)} \\ \mu_m^{(y)} \end{bmatrix} \quad \Sigma_m^{(z)} = \begin{bmatrix} \Sigma_m^{(xx)} & \Sigma_m^{(xy)} \\ \Sigma_m^{yx} & \Sigma_m^{(yy)} \end{bmatrix} \tag{2}$$

Given x_t, the estimated target vector $\hat{y}_t$ is determined by as follows:

$$\begin{aligned} \hat{y}_t &= E[y_t|x_t] \\ &= \sum_{m=1}^{M} p(m|x_t, \lambda^{(z)}) E_{m,t}^{(y)} \end{aligned}, \tag{3}$$

where

$$p(m|x_t, \lambda^{(z)}) = \frac{w_m N(x_t; \mu_m^{(x)}, \Sigma_m^{(xx)})}{\sum_{n=1}^{M} w_n \cdot N(x_t; \mu_n^{(x)}, \Sigma_n^{(xx)})}, \tag{4}$$

$$E_{m,t}^{(y)} = \mu_m^{(y)} + \Sigma_m^{yx} \Sigma_m^{(xx)^{-1}} (x_t - \mu_m^{(x)}). \tag{5}$$

2.2. Proposed conversion method

The conventional method is reasonably effective. However, the performance is of the conversion is still insufficient, discontinuities often appear in the sequences of the estimated target vectors. The reason is that the conversion is independently performed at individual frames. The correlations between frames are ignored (see Figure 1(a)). In fact, due to the finite velocity of lip motions, current visual feature depends on not only current audio feature but also its previous visual features. The visual feature sequence resembles a Markov chain shown in Figure 1(b). However, it is a special Markov chain with a "knob" (the audio feature) to control the transition probability.

Let y_t be the current visual feature and y_t^p represent its previous visual feature state. If y_t^p is always available, we can use a GMM to model the joint probability density of x_t, y_t^p and y_t

$$P(Z_t|\lambda^{(Z)}) = \sum_{m=1}^{M} w_m \cdot N(Z_t\,; \mu_m^{(Z)}, \Sigma_m^{(Z)}). \tag{6}$$

where $Z_t = [X_t\,; y_t]$, $X_t = [x_t; y_t^p\,]$.

Given x_t, the transition probability density is as follows:

$$P(y_t|y_t^p, x_t, \lambda^{(Z)}) = \sum_{m=1}^{M} P(m|X_t, \lambda^{(Z)}) P(y_t|X_t, m, \lambda^{(Z)}) \tag{7}$$

$p(m|X_t, \lambda^{(Z)})$ has the same form as that in equation 4 except that x_t is replaced by X_t. $P(y_t|X_t, m, \lambda^{(Z)})$ is a normal distribution with mean vector $E_{m,t}^{(y)}$ and covariance matrix $D_{m,t}^{(y)}$,

$$E_{m,t}^{(y)} = \mu_m^{(y)} + \Sigma_m^{yX} \Sigma_m^{(XX)^{-1}} (X_t - \mu_m^{(X)}). \tag{8}$$

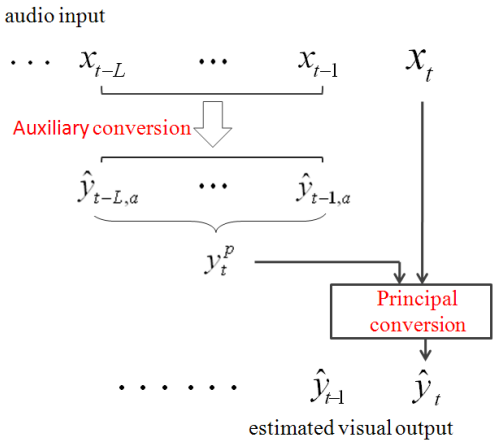

Fig. 2: Overview of the proposed conversion method. It consists of a principal conversion and an auxiliary conversion.

By using MMSE estimation, the estimated visual vector $\hat{y}_t$ is determined as

$$\begin{aligned} \hat{y}_t &= E[y_t|X_t] \\ &= \sum_{m=1}^{M} p(m|X_t, \lambda^{(Z)}) E_{m,t}^{(y)} \end{aligned} \tag{9}$$

Then, the problem becomes finding a proper representation of the previous visual feature state y_t^p. We can simply use y_{t-1} as the previous visual feature state of y_t, i.e.

$$y_t^p = y_{t-1} \tag{10}$$

Our experiments show that if the ground truth value of y_{t-1} is used, the estimated vector $\hat{y}_t$ is almost the same as its ground truth. However, the ground truth value of y_{t-1} is available during GMM training, but it is unavailable during practical conversion. An alternative is to use $\hat{y}_{t-1}$ as an approximate of y_{t-1}. Given an initial value, the visual feature vector sequence can be estimated in an iteratively way. The problem is that the conversion error generated in each frame would accumulate. It doesn't necessarily lead to reasonable converted results.

To solve the problem, we incorporate an auxiliary audio-to-visual conversion. The overview of the proposed method is shown in Figure 2. Our method consists of a principal conversion and an auxiliary conversion, The conversion described in equation 9 is referred as principal conversion. Since the auxiliary conversion deals with those audio feature vectors in the past, any direct audio-to-visual conversion method can be used, including affine transformation [16], artificial neural networks and trajectory-based GMM [14]. Here we use the conversional GMM-based conversion as the auxiliary conversion, its number of mixture component is denoted as M_a, and the estimated visual vector is denoted as $\hat{y}_{t,a}$.

The conversion procedure is as follows. When the current visual feature vector x_t becomes available, $\hat{y}_{t-L,a}, \cdots, \hat{y}_{t-2,a}$ and $\hat{y}_{t-1,a}$ are obtained by using the auxiliary conversion. Next, the previous visual feature state is calculated as follows

$$y_t^p = \frac{1}{L} \sum_{i=1}^{L} \hat{y}_{t-i,a} \tag{11}$$

L is the number of frames used to calculate the average. Finally, $\hat{y}_t$ is calculated using equation 9. Since the update of y_t^p is independent of $\hat{y}_t$, conversion error will not accumulate. Moreover, y_t^p is available

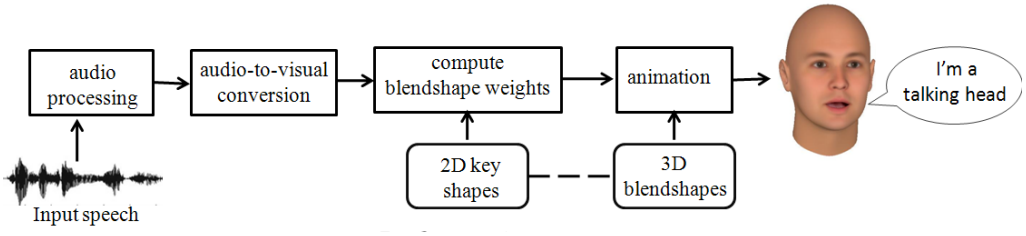

Fig. 3: Overview of the online processing pipeline.

both in training and in conversion. Although y_t^p is not the ground truth values of y_{t-1}, it does represent the main trends of the previous visual feature vectors, the estimated target vector is forced to follow the main trends.

Audio-visual model training. The training procedure of the two GMMs is as follows. First, the auxiliary GMM is trained using training set $\{x_t, y_t\}$. Then, the converted results $\hat{y}_{t,a}$ are calculated for all samples in the training set using equation 3, and all y_t^p are calculated using equation 11. In this way, another training set $\{x_t, y_t^p, y_t\}$ is obtained. Finally, this training set is used to train the principal GMM. We train the GMMs using the EM algorithm.

Collecting audio-visual training data. A human subject is first recorded using a video camera as he/she utters a predetermined speech corpus [17]. Given the recorded videos and audio, training data are collected as follows. For each video image, we track 68 facial feature points. The 2D positions of the 68 feature points are concatenated to form a shape vector s. Principal component analysis (PCA) is applied to the shape vector, and the resulting PCA coefficients are used as visual features, so we also refer to y_t as PCA coefficients. For each audio frame, Mel-Frequency Cepstral Coefficients (MFCC) are extracted and adopted as audio features.

3. ANIMATION SYNTHESIS

3.1. System overview

An overview of the online processing pipeline is shown in Figure 3. Input to the system is the recorded speech, audio features (MFCC feature vectors) are calculated and fed into the audio-visual converter, resulting in visual PCA coefficients. Since we make facial animation using a blendshape model, these PCA coefficients need to be transformed into blendshape weights to drive the virtual character.

3.2. Compute blendshape weights

Blendshape models are very popular in facial animation [18, 19]. For a blendshape model, animations can be generated by shape morphing or through linear combinations of basis poses.

Similar to [18], we use 39 blendshapes in our examples. Let $\mathrm{b} = [\mathrm{b}_1, \mathrm{b}_2, \cdots, \mathrm{b}_{39}]$ be the blendshape weight vector. To compute blendshape weights, we need to create 39 2D key shapes that are in one-to-one correspondence with those 3D blendshapes. These 2D key shapes are created as follows. First, we select a 2D key shape corresponding to neutral face in the training images. Then, we manually define a corresponding model point on the 3D face model for each feature point of the 2D key shape. Given a 3D blendshape, its deformation can be transferred to the 2D key shape by using an MEPG-4 based animation framework [20]. In this way, we obtain 39 2D key shapes $\mathbf{K} = [K_1, K_2, \cdots, K_{39}]$. We assume that these 2D key shapes share the same weights with those 3D blendshpaes [21].

Given the PCA coefficients y_t, the face shape vector s_t can be reconstructed using principal components. Then the blendshape weights are solved by minimizing the following energy

$$E_t = \left\| \sum_{j=1}^{39} b_j K_j - s \right\|^2 + \beta \cdot \sum_{j=1}^{39} b_j^2 \tag{12}$$

where β is a constant, and is set to 10 in our experiments. Since $b_j \in [0, 1]$, by letting

$$b_j = \frac{1}{1 + \mathrm{e}^{-r \cdot \theta_j}} \tag{13}$$

we can solve for θ_j instead of b_j, the constant r is set to 0.2 in our experiments. Then, it becomes a unconstrained minimization problem, and can be solved using an iterative gradient solver [22]. The gradient is calculated as follows

$$\frac{\partial E_t}{\partial \theta} = 2 \cdot G \cdot \mathbf{K}^T (\mathbf{K} \cdot b - s) + 2\beta \cdot G \cdot b \tag{14}$$

where $\theta = [\theta_1, \cdots, \theta_{39}]^T$, G is a diagonal matrix with diagonal entries equal to $\frac{\partial b_j}{\partial \theta_j}$.

4. EXPERIMENTAL RESULTS

4.1. Resulting animations

We use the LIPS 2008 Visual Speech Synthesis database [17] as our audio-visual corpus. This database has 278 video files with corresponding audio tracks, each being one English sentence spoken by a single native speaker with neutral emotion. The video frame rate is 50 fps. For each video image, we use Active Shape Model [23] to locate 68 facial features, 10-dimensional PCA Coefficients associated with each frame are adopted as visual features. For audio tracks, MFCC vectors are extracted with a 20ms time window shifted every 10ms. The visual features are interpolated up to the same frame rate as the MFCC vectors. Then, the audio-visual feature vectors are used to train our audio-visual model.

At run time, given novel speech signals, MFCC vectors are calculated and then mapped to PCA coefficient vectors. Finally, the blendshape weight vectors are computed and used to drive the digital character. We implemented our system on a PC with an Intel Core 2 Duo (2.8 GHz) CPU. The system runs in real time at 22 fps. Fig 4 shows a few synthesized animations.

For more results, please refer to the supplementary video. (http://home.ustc.edu.cn/%7Eluocw/animation2.wmv).

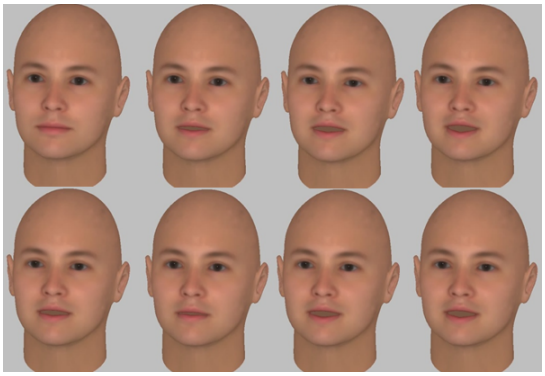

Fig. 4: Snapshots from the synthesized speech animation.

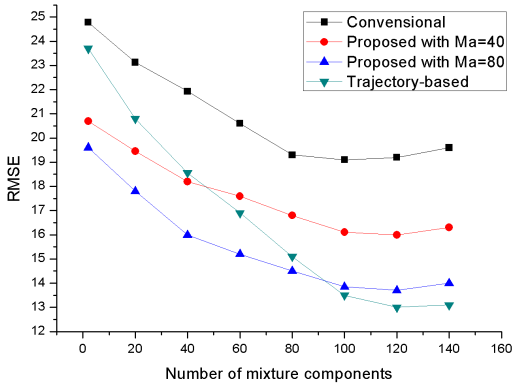

Fig. 5: The RMSE as a function of the number of mixture components.

4.2. Evaluation

The performance of audio-to-visual conversion is objectively evaluated by the root-mean-square error (RMSE). It is calculated for the difference between the measured and the estimated PCA coefficient vectors, i.e.

$$RMSE = \sqrt{\frac{1}{n} \cdot \sum_{t=1}^{n} ||\hat{y}_t - y_t||^2} \tag{15}$$

A 1/5 cross-validation test is conducted to measure the accuracy of the conversion under open conditions. The training samples are divided into 5 partitions, and then one of the partitions is reserved for testing by turns, while other partitions are used for training. Finally, the average RMSE is calculated.

We compare the proposed method with the conventional and the trajectory-based method [14] under different conditions. In each case, the number of mixture components is varied from 1 to 140. For the proposed conversion, eleven frames are used to calculate the previous visual feature state, i.e. L=11. M_a is set to 40 and 80, respectively. The RMSE as a function of the number of mixture components is shown in Figure 5. Figure 5 shows that the RMSE is greatly reduced after considering the previous features. The reason is that the estimated visual vectors are constrained by the previous visual states, many sudden jumps are alleviated. When the number of mixture components is small, our method outperforms the trajectory-based method. With increasing number of mixture components, the trajectory-based method slightly outperforms ours. This is probably because the dynamic features have been used in the trajectory-based method, it needs more mixture components to model the joint space of both static and dynamic features.

Since speech animation is to provide a natural human-machine communication method, subjective evaluations from human observers are more appropriate than objective measurements. A subjective mean opinion scoring test was also carried out to compare the conventional, the trajectory-based and the proposed audio-to-visual conversion methods. We randomly select 6 sentences from the LIPS 2008 database. Then we synthesized 6 facial animation videos for each of these three methods. Each video also includes the ground truth input speech audio. Ten volunteers are required to score the animations in two facets: smoothness and audio-visual consistency from 1 (worst) to 5 (best). The results of the scoring are shown in Table 1. It is shown that the scores for the proposed method is much higher than the conventional GMM based method. Again, the trajectory-based method is slightly superior to ours. However, our method converts audio to visual in real time while the trajectory-based method has a latency time no less than the length of one utterance. Thus, our method is more suitable for real-time system.

Table 1: The averaged subjective scores for smoothness and audio-visual consistency

	conventional	proposed	trajectory based
smoothness	3.22	4.15	4.28
consistency	3.01	3.93	4.06

5. CONCLUSIONS

We propose a real-time audio-to-visual conversion method based on GMM. We assume that the current visual feature depends on not only the speech signal but also its previous visual features, and then the visual feature sequence behaves as a Markov chain. A GMM is used to model the transition probability density. Since the ground truth values of the previous visual feature states are unavailable during conversion, we incorporate an auxiliary conversion which provides previous visual feature states for the principal conversion. After audio-to-visual conversion, the visual features are transformed into blendshape weights to synthesize speech animations. The output speech animations are quite realistic, and the synthetic lip motions synchronize well with the speech.

6. RELATION TO PRIOR WORK

Our work focuses on speech-driven facial animation synthesis. For audio-to-visual conversion, we incorporate previous visual features into the conversion to avoid sudden jumps of estimated visual features, while the earlier work in [15] only uses current audio feature. To compute blendshape weights, Cao et al. [19] make use of manually created animations to define animation priors, this procedure is very exhausting. We directly add a regularized item and also obtain plausible blendshape weights.

7. REFERENCES

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

[2] J. Ostermann and A.Weissenfeld, "Talking faces-technologies and applications," in *International conference on pattern recognition*, 2004.

[3] M. Brand, "voice puppetry," in *Proceedings SIGGRAPH 1999*, 1999.

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

[5] T. Ezzat, G. Geiger, and T. Poggio, "Trainable video realistic speech animation," in *Proc. ACM SIGGRAPH2002*, 2002.

[6] L. Wang, X. Qian, W. Han, and F. Soong, "Synthesizing photo-real talking head via trajectory-guided sample selection," in *interspeech 2010*, 2010.

[7] Z. Deng, U. Neumann, J. Lewis, T. Kim, M. Bulut, and S. Narayanan, "Expressive facial animation synthesis by learning speech coarticulation and expression spaces," *IEEE Trans. on visualization and computer graphics*, vol. 12, pp. 1523–1534, 2006.

[8] L. Wang, W. Han, and F. Soong, "High quality lip-sync animation for 3d photo-realistic talking head," in *ICASSP 2012 Proceedings*, 2012.

[9] G. Hofer, J. Yamagishi, and H. Shimodaira, "Speech-driven lip motion generation with a trajectory hmm," in *Interspeech 2008*, 2008.

[10] S. Fu, R. Gutierrez-Osuna, A. Esposito, P. Kakumanu, and O. N. Garcia, "Audio/visual mapping with cross-modal hidden markov models," *IEEE Transactions on Multimedia*, vol. 7, pp. 243–252, 2005.

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

[12] 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.

[13] T. Toda, A. W. Black, and Tokuda, "Statistical mapping between articulatory movements and acoustic spectrum using a gaussian mixture model," *Speech Communication*, vol. 50, pp. 215–227, 2008.

[14] T. Toda, A. Black, and K. Tokuda, "Voice conversion based on maximumlikelihood estimation of spectral parameter trajectory," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, pp. 2222–2235, 2007.

[15] Y. Stylianou, O. Cappe, and E. Moulines, "Continuous probabilistic transform for voice conversion," *IEEE Trans. Speech and Audio Processing*, vol. 6, pp. 131–42, 1998.

[16] H. Yehia, P. Rubin, and E. Vatikiotis-Bateson, "Quantitative association of vocal-tract and facial behavior," *Speech Communication*, vol. 26, pp. 23–43, 1998.

[17] B. Theobald, S. Fagel, G. Bailly, and F. Elisei, "Lips2008: Visual speech synthesis challenge," in *Proceedings of Interspeech*, 2008.

[18] T. Weise, S. Bouaziz, H. Li, and M. Pauly, "Realtime performance-based facial animation," in *Proceedings SIGGRAPH 2011*, 2011.

[19] C. Cao, Y. Weng, S. Lin, and K. Zhou, "3d shape regression for real-time facial animation," in *Proceedings SIGGRAPH 2013*, 2013.

[20] A. Tekalp and J. Ostermann, "Face and 2-d mesh animation in mpeg-4," *Signal Processing: Image Communication*, vol. 15, pp. 387–421, 2000.

[21] E. Chuang and C. Bregler, "performance driven facial animation using blendshape interpolation," Stanford University, Tech. Rep., 2002.

[22] D. Liu and J. Nocedal, "On the limited memory bfgs method for large scale optimization," *Mathematical Programming*, vol. 45(1-3), pp. 503–528, 1989.

[23] T. Cootes and C. Taylor, "Statistical models of appearance for computer vision," The University of Manchester School of Medicine, Tech. Rep., 2004.