Lip Animation Synthesis: a Unified Framework for Speaking and Laughing Virtual Agent

Yu Ding, Catherine Pelachaud

CNRS-LTCI, Télécom-ParisTech, Paris, France

{yu.ding, catherine.pelachaud}@telecom-paristech.fr

Abstract

This paper proposes a unified statistical framework to synthesize speaking and laughing lip animations for virtual agents in real time. Our lip animation synthesis model takes as input the decomposition of a spoken text into phonemes as well as their duration. Our model can be used with synthesized speech. First, Gaussian mixture models (GMMs), called $lip\ shape\ GMMs$, are used to model the relationship between phoneme duration and lip shape from human motion capture data; then an interpolation function is learnt from human motion capture data, which is based on hidden Markov models (HMMs), called $HMMs\ interpolation$. In the synthesis step, $lip\ shape\ GMMs$ are used to infer a first lip shape stream from the inputs; then this lip shape stream is smoothed by the learnt $HMMs\ interpolation$, to obtain the synthesized lip animation. The effectiveness of the proposed framework is confirmed in the objective evaluation.

Index Terms: lip animation, speech to animation, interactive virtual agent, laughter, speech, Gaussian mixture models (GMMs), hidden Markov models (HMMs)

1. Introduction

Interactive virtual agents (IVAs) are human-like virtual characters. IVAs systems use speech synthesizer to output what the agent says. They can also express their intentions, attitudes and emotions as humans do through nonverbal behaviors. They are used in various applications of human-computer interaction, such as web assistants and information providers, or as NPC in video games.

Humans are very skilled at reading facial expression and lip motion of human [1] as well as of IVAs [2, 3, 4, 5]. In particular, humans notice when lip animation and speech are not fully matched and synchronized [1, 6]. Lip animation is not built from a succession of the lip shape of phonemes but ought to capture co-articulation effect [7]. Lip animation for speech has been studied quite a lot and several challenges were conducted [8, 9].

The model of Cohen and Massaro [7], one of the first models of lip synthesis, uses dominance functions to model and infer co-articulation influences between neighboring phonemes. Similar approaches also considered the influence of surrounding phonemes [10, 11, 12]. Such works used specific functions to define the co-articulation relationship of specific neighboring phonemes. However it is time consuming to cover all the possible combinations of neighboring phonemes. These approaches are not easily manipulated to new speakers or other languages. Other works [13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 5, 23, 24, 25, 26, 27, 28] used statistical models to learn the co-articulation relationships, where specific combinations

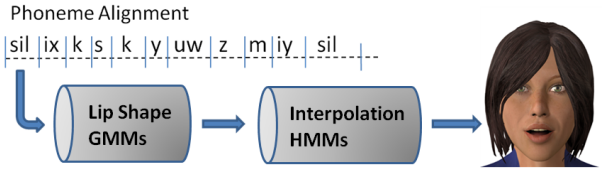

Figure 1: *Overview of synthesis framework.*

of neighboring phonemes are not explicitly specified. Such statistical models can be learned for a new speaker. Based on statistical model, HMM-based synthesis framework is a classical approach to simulate speech lip animation synthesis [14, 15, 29, 16, 20, 21, 23, 24, 25, 26, 28]. Usually, an HMM is built for each phoneme in context, that is taking into account the vowel and/or consonant that surround the phoneme. During the synthesis step, one concatenates the context-dependent phoneme-sized HMM according to the input sequence of phonemes and of their duration; then one can obtain a sequence of HMM states. In a second step, given a state sequence, the parameter generation algorithm [30] is used to synthesize a smooth animation of the lip shapes.

Such a synthesis approach makes the assumption that lip shape is characterized only by phoneme type and not also by its duration. However, lip shape depends not only on phoneme type but also on phoneme duration [7, 31]. For example, for short phoneme duration lip shape may not reach its standard position [31]. This temporal information is ignored by HMM-based synthesis framework. Furthermore, the synthesis generation algorithm proposed by [30] cannot synthesize animation in real-time.

While many research works have focused on audio-visual speech synthesis during neutral speech, other researchers have developed models for emotional speech [19] [32] and for laughter [33] [34]. These last works on laughter use Hidden Markov Models (HMMs) but do not work on real-time. Ding et al. [35] proposed linear regression models to infer laughter lip animation in real-time. However, no model has been proposed to synthesize both speech and laughter lip animation.

Existing lip animation models can be classified as text-driven or speech-driven approaches. Although these both approaches use different input signals, they both use phoneme information. The former uses the text decomposed into phonemes stream; the latter use the stream of acoustic features, which is strictly linked to phoneme. Agents interact in real-time with human users. Their speech is obtained by using speech synthesizer technology. Therefore, text-driven approaches are used in most applications of IVAs [10, 11, 12, 7].

We aim to develop a unified statistical model to infer speech

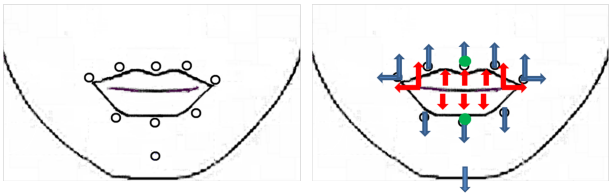

Figure 2: *Left figure: Position of motion capture sensors on lip and jaw. Each sensor can capture 3-dimensional motion data. Right figure: Lip and jaw facial animation parameters (FAPs): 22 FAPs are defined for the lip and 1 FAP for the jaw. FAPs are represented by arrows and circles. The 11 blue arrows are outer lip FAPs and jaw FAP; the 8 red arrows are inner lip FAPs; the green circles present forward and backward displacements of the outer lip.*

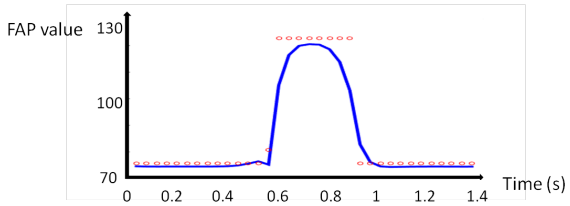

Figure 3: *Example of jaw outputs by lip shape $GMMs$ and $HMMs$ interpolation. The spoken sentence is "yeah". The red circles are computed by lip shape $GMMs$; the blue curve is obtained by $HMMs$ interpolation. During the synthesis step, lip shape $GMMs$ determine the sequence of lip shapes for each time frame; $HMMs$ interpolation is used to smooth the sequence of lip shapes.*

and laughter lip shape from speech/laughter text information. The statistical model captures co-articulation effects and runs in real-time. In our work, speech and laughter lip animations are independent with each other; computing lip animation synthesis for speaking while laughing is beyond the scope of this paper. First, the built GMMs, called *lip shape* $GMMs$, are used to infer lip shape for each phoneme. Secondly, the built HMMs, called $HMMs$ *interpolation*, are used as an interpolation function, taking the inferred lip shape stream as input, and then generating smooth lip animation. Figure 1 shows the overview of our synthesis framework.

In the next section 2 we detail our model. Then we describe the objective evaluation we conducted. Finally we conclude our paper.

2. Real-Time Lip Animation

This section introduces the statistical framework of our real-time lip animation synthesis. As mentioned above, the statistical framework for speech and laughter is identical; that is both take as input a list of phonemes and their duration information. The output features are identical in speech and laughter, which be used to animate IVAs. Speech and laughter animation models are independent with each other.

Our speech lip animation framework is built on a dataset called IEMOCAP [36]. 169 episodes are used in our work. Each episode contains speech data as well as lip and jaw motions. Lip and jaw motions are recorded through 9 motion capture markers capturing point displacement in 3D (giving a total of 24 dimensions for lip and 3 dimensions for jaw). Figure 2 shows the layout of lip motion capture markers. More details about dataset can been found in [36].

The facial animation model of our agent model follows the MEPG-4 standard [37]. In MEPG-4 standard, 22 lip facial animation parameters (FAPs) and 1 jaw FAP are defined to describe all the possible lip shapes (specifying inner and outer lip shapes) and jaw position. Figure 2 illustrates lip and jaw FAPs. Jaw FAP can be obtained directly from jaw motion capture marker. To define a mapping between lip dimension from the mocap data and the FAPs, we can notice that the 24 dimensions obtained by the mocap data, only 12 correspond to the outer lip parameters FAPs of the virtual agent. For the other 10 FAPs defining the inner lip shapes, there is no direct correspondence. The IEMOCAP did not capture this information. For sake of simplicity, in our model, inner lip moves similarly as the outer lip. Such a simplification was made in [38, 39]. Thus, 13 motion features are taken into account including 12 lip FAPs and 1 jaw FAP.

The IEMOCAP dataset [36] provides phonemes information, including phonemes label and their duration for each episode, which was obtained from Ubiqus [40]. In total, 48 phonemes labels are considered.

Laughter lip animation synthesis is built on a dataset called AVLaughterCycle [41]. The database contains human laughter audio and video. The lip motion is detected by an open-source face tracking tool - FaceTracker [42]. The tracking algorithm outputs value for the outer lip and jaw parameters; having these values, we extrapolate the value of the inner lip shapes as done for speech inner lip FAPs.

In building laughter animation synthesis framework, laughter phoneme is used in reference to (speech) phoneme. Laughter phoneme is defined by [43], which categorized laughter audio into 12 laughter phonemes according to human hearing perception.

Our algorithm is made of two steps. It works as follows. At first, *lip shape* $GMMs$ are used to model the relationship between phoneme duration and visemes (phoneme lip shape) (described in Section 2.1). Then, an interpolation function ($GMMs$ *interpolation*) is built by learning human lip motion (detailed in Section 2.2). Finally, the real-time synthesis is explained in Section 2.3. That is, *lip shape* $GMMs$ is used to estimate viseme stream corresponding to the input phoneme sequence. In this step, the viseme stream is defined by a unique position that runs over for the whole phoneme. Then such viseme stream is smoothed by $HMMs$ *interpolation*. Figure3 illustrates the outputs of *lip shape* $GMMs$ and $HMMs$ *interpolation*. The framework overview is illustrated in Figure1.

2.1. Modeling Lip Shape

Lip shape depends not only on phoneme type by also on phoneme duration [31]. This section introduces how to build *lip shape* $GMMs$, which is used to capture the relationship between phoneme duration and visemes. In the following we present how we build the training dataset that we used to train *lip shape* $GMMs$. Then, we describe how to infer viseme according to the given phoneme and its duration; this step is used in the synthesis part.

We compute how many times each of them occurs in all the training episodes. Let us call N the number of occurrence of phoneme label pho; each phoneme can occur several times in one episode. We compute the envelop of the WAV stream. We segment this envelop for the phonemes sequence. For each

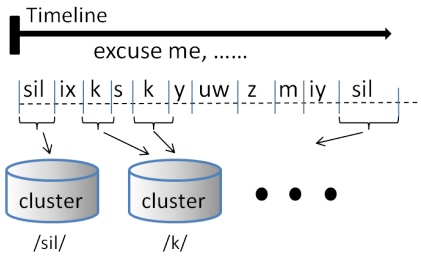

Figure 4: *Building datasets for each phoneme*. In total, 48 datasets are built. One dataset corresponds to one phoneme. Each dataset contains N samples. N being the number of phoneme occurrences. Each sample consists of 12-dimensional motion vector and 1-dimensional phoneme duration.

phoneme segment, we consider the maximum point on the envelop. The apex of a viseme is defined by the temporal value of the maximum of the phoneme segment. We collect the apex of the 13 labial parameters of each occurrence of each phoneme pho and build motion sets (see Figure 4), ${m_i}^{pho}$, $i = 1, ..., N$. m_i^{pho} is 13-dimensional vector composed of the 13 FAPs. Each occurring phoneme is also characterized by its duration, d_i^{pho}. For each phoneme label, we have a set of ${m_i}^{pho}$ and d_i^{pho}. This set is noted by md^{pho}.

2.1.1. Training: Building $GMMs_{m,d}$

N $GMMs_{m,d}$ are built (where N is equal to 48 for speech lip and 12 for laughter lip). Each $GMMs_{m,d}$ of N is used to model the joint probability density of m_i^{pho} and d_i^{pho}, which is recorded in md^{pho}. The $GMMs_{m,d}$ for phoneme pho is described as follows:

$$P(md_i^{pho}|\lambda^{(md)}) = \sum_{c=1}^{C} \alpha_c N(md_i^{pho}; \mu_c^{(md)}, \sigma_c^{(md)}) \quad (1)$$

where $md_i^{pho} = [{m_i^{pho}}^T \; {d_i^{pho}}^T]^T$. The notation ^{T} denotes the transpose of the vector. The parameter set of the GMM is $\lambda^{(md)}$. The mixture component index is c. The total number of mixture components is C (where C is equal to 5). The weight of the c^{th} mixture component is α_c. The distribution with mean μ and covariance σ is denoted as $N(.; \mu, \sigma)$. The mean vector $\mu_c^{(md)}$ is a 14-dimensional joint vector and the corresponding covariance matrix is $\Sigma_c^{(md)}$. The 14-dimensional mean vector $\mu_c^{(md)}$ is composed of 14-dimensional mean vector of motion and $\mu_c^{(m)}$, 1-dimensional mean of duration, $\mu_c^{(d)}$.

As described above, N $GMMs_{m,d}$ can be separately built for N phonemes.

2.1.2. Synthesis: Extracting $GMMs_d$ and $GMMs_m$

In the synthesis step, $GMMs_{m,d}^{pho}$ can be decomposed into $GMMs_d^{pho}$ and $GMMs_m^{pho}$, by respectively ignoring either the element of motion or the element of phoneme duration from $\mu_m^{(md)}$ and $\Sigma_m^{(md)}$. In $GMMs_m^{pho}$ mean vector, $\mu_m^{(m)}$, is a 13-dimensional vector while in $GMMs_d^{pho}$ mean vector, $\mu_m^{(d)}$, is 1-dimensional vector. Such a framework based on $GMMs$ is called *lip shape* $GMMs$.

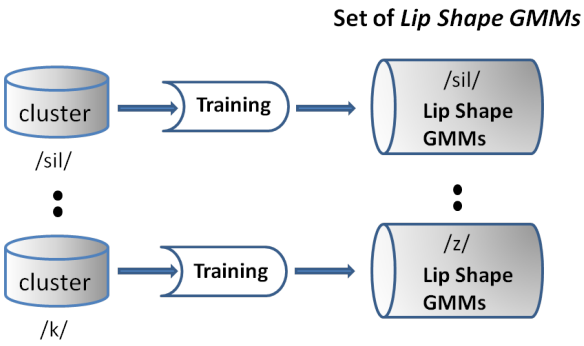

Figure 5: *Training lip shape $GMMs$ for each phoneme.*

2.1.3. Synthesis: Inferring lip shape sequence

Given the input of phonemes sequence, we can align a sequence of $GMMs_d^{pho}$ according to phoneme label. Then each $GMMs_d^{pho}$ is used to synthesize the segment of lip animation. The following introduces how to synthesize lip animation from a given $GMMs_d^{pho}$.

Given the phoneme duration, d, $GMMs_d^{pho}$ is used to select the most likely component as follows:

$$c = \underset{c=1,...,C}{\operatorname{argmax}} \alpha_c P(d|\mu_c, \sigma_c) \quad (2)$$

where the c-th component is determined as the occurring component. Then the occurring component is applied to $GMMs_m^{pho}$.

Once the occurring component is determined, the mean vector, μ_c, of the c-th component in $GMMs_m^{pho}$ is viewed as the lip shape output, m. Then m is repeated d times; that is, the same lip shape output is used for the whole duration, d.

As introduced above, the GMMs-based framework, *lip shape* $GMMs$, is used to learn the relationship between lip shape and phoneme defined by its label and its duration. Then this framework is used to infer lip shape for all phonemes at each time step. In this step, the output lip shapes are viewed as being constant over the whole phoneme and are independent from the neighboring phonemes. Figure 3 shows the output lip shapes (displayed as red circles) as obtained from *lip shape* $GMMs$.

2.2. Training an Interpolation Function

In the previous section 2.1, we introduce how to infer lip shape for each phoneme. In the following we detail how to smooth lip shapes using neighboring phonemes, thus capturing some form of co-articulation effect.

2.2.1. Training: Building $HMMs$ interpolation

A 13-dimensional motion stream, m^l, $l = 1, ..., 13$, can be obtained from each training episode. The speed feature, Δm^l, and the acceleration, $\Delta\Delta m^l$, are calculated. The motion joint vector is defined as $o^l = [{m^l}^T, {\Delta m^l}^T {\Delta\Delta m^l}^T]^T$. All the motion joint vectors, o^l, are grouped as a set, $\{o^l\}$. Each set, $\{o^l\}$, is clustered by the LBG algorithm [44]. We obtain a codebook with Q representative subsets (where Q is equal to 50). 13 codebooks of size Q are separately built on one-dimensional motion stream. Each frame of the motion data is clustered into a subset. In each subset, the mean and the covariance are calculated to

characterize this subset. Q pairs of mean and covariance can be obtained for each dimensional motion stream and they are used to define the emission probabilities of HMM. The state transition probabilities are learnt by computing the transition number. As described above, 13 $HMMs_{o^l}$ $interpolation$ can be built separately for 13-dimensional motion.

2.2.2. *Synthesis: Smoothing the lip shape stream*

During the synthesis step, one HMM_{m^l} with observation, m^l, can be extracted from the trained HMM_{o^l} with observation, o^l, by ignoring the observations Δm^l and $\Delta\Delta m^l$.

As introduced in 2.1, $lip\ shape\ GMMs$ can produce lip shape stream corresponding to several phonemes. Lip shape stream is composed of m^l, $l = 1, ..., 13$. Lip shape is constant over one phoneme. Given lip shape stream, a state sequence, q, can be determined from HMM_{m^l} by using Viterbi algorithm.

q is applied to GMM_{o^l}. Given $GMMs_{o^l}$ and a state sequence, q, a smooth trajectory, m^l, can be synthesized directly by solving the equation as follows:

$$\frac{\partial log P(Wm^l|q, HMMs_{o^l})}{\partial m^l} = 0 \tag{3}$$

where W the operation matrix to satisfy $o^l = Wm^l$. This equation has been solved by Tokuda et al. [30].

As introduced above, HMMs play the role of interpolation function. For the sake of simplicity, such HMMs are called $interpolation\ HMMs$. First, the rough observation stream (constant over one phoneme) is used as input to determine a state sequence. Then, from this state sequence, a smoothing motion stream is generated as output.

2.3. Real-Time Lip Animation Synthesis

Section 2.1 presented how to infer lip shape at one time step, based on $lip\ shape\ GMMs$; Section 2.2 explained the HMMs-based interpolation function ($HMMs\ interpolation$). Generally, once lip shape stream is inferred by $lip\ shape\ GMMs$; it can be smoothed by $HMMs\ interpolation$. The following details how to perform the lip animation synthesis in real time using $lip\ shape\ GMMs$ and $HMMs\ interpolation$.

Given the input phonemes stream, vowels (V) are used to segment phonemes stream as unit, which is inspired by the definition of syllables as the fundamental units of speech production [7, 31, 45]. The last phoneme in one unit is the first phoneme in the next unit. The unit can be group of V-V, V-Consonant-V or V-Consonants-V. Three consonants (p, b, m) define labial closure and they are also used to segment phonemes stream as vowels do. In the step of synthesis, each unit is used sequentially as input to the synthesis framework.

3. Results and Evaluation

Figure 6 shows two trajectory examples. To perform objective evaluation of our model, we compute the difference (i.e. error rate) in lip shape between the original lip movement and the synthesized one. To understand where errors may be more prominent, we compute the differences in lip shape for specific features. We rely on works done in phonetics studies that described lip shapes by 4 labial parameters [7]:

1. lip openness: distance between top and bottom middle lip positions.
2. lip extension: distance between left and right lip corners.
3. top lip protrusion.

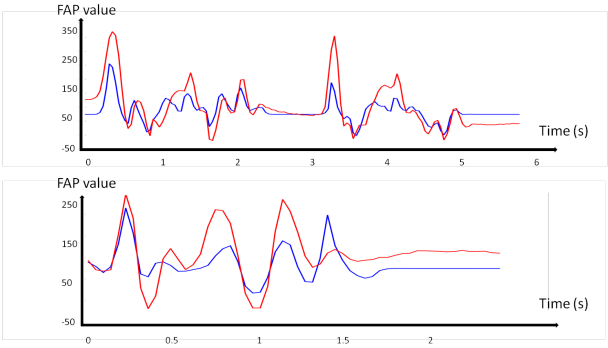

Figure 6: *Two Trajectory Examples. The red curve corresponds to the human jaw trajectory; the blue one is the synthesized jaw trajectory.*

4. bottom lip protrusion.

These 4 labial parameters are taken into account to evaluate objectively the performance of our model. We calculate the differences in terms of 4 labial parameters between the original and the synthesized motions, as follows:

$$diff = \frac{Ori - Syn}{MaximumValue} \tag{4}$$

where $diff$ is the resulting difference; Ori is the value from human data; Syn is the synthesized value and $MaximumValue$ is the maximum value of the 4 labial parameters calculated on all the training episodes. Hence, $diff$ ranges from 0 to 1.

We separate the whole speech or laughter database in 80% for training and 20% for testing. We run our algorithm on this 20%. We compute the value of the 4 phonetic parameters from the synthesized lip animation. We compute the difference between the value of these 4 parameters for the synthesized animation and for the natural speech. We repeat this computation 20 times; that is, for 20 times, we divide the database into 2 sets, 80% for training and 20% for testing and compute the error rate for the lip displacement. We take the average of the 20 error rates and of the standard deviation for computing the final error rate. Figure 6 shows two trajectory examples. The segment trajectories corresponding to the phoneme *silence* for speech and for laughter are not taken into account in our objective evaluation.

The experimental results on speech lip animation are showed in Table 1. 169 speech episodes are used in total. The results show that our statistical framework outperforms the HMM-based synthesis framework [29, 30] trained by Maximum Likelihood Estimation (MLE). The results by HMM-based synthesis framework is based on the same dataset.

In HMM-based synthesis framework, all the inferred values at each frame time of one animation sequence are dependent with each other; that is, with HMM-based synthesis model, co-articulation occurs between both, close and distant neighboring phonemes. This may create artifact and may lower lip animation quality as studies have shown that co-articulation has a range of influence, even if this range varies depending on the phoneme sequences [7, 31]. Furthermore, HMM-based synthesis framework do not work in real-time while our model does.

Our real-time framework emphasizes not only the importance of phoneme on lip shape with the $lip\ shape\ GMMs$, but also it takes into account co-articulation of close neighboring

	Proposed model	HMM-based framework[29]
lip openness	0.049 (0.029)	0.19 (0.056)
lip extension	0.061 (0.041)	0.23 (0.064)
top lip protrusion	0.163 (0.055)	0.34 (0.071)
bottom lip protrusion	0.132 (0.047)	0.31 (0.031)

Table 1: Speech Lip Results ($diff$): Performance of the models with respect to the synthesis quality. Performances are averaged results gained on 20 experiments (standard deviations are given in brackets). $diff$ is in the range of 0 and 1

	Proposed model	regression model[35]
lip openness	0.18 (0.076)	0.22 (0.067)
lip extension	0.19 (0.039)	0.25 (0.059)
top lip protrusion	0.21 (0.061)	0.26 (0.024)
bottom lip protrusion	0.27 (0.059)	0.31 (0.041)

Table 2: Laughter Lip Results: Performance of the models with respect to the synthesis quality. Performances are averaged results gained running 20 experiments (standard deviations are given in brackets).

phonemes with the $HMMs\ interpolation$. The $lip\ shape\ GMMs$ is capable of modeling viseme adaption depending on phoneme duration; thus, it simulates how a viseme may not reach its position if phoneme duration may not allow it [31].

Our proposed $HMMs\ interpolation$ is inspired by the parameter synthesis algorithm [30], which is used directly in HMM-based synthesis framework [15, 29] to smooth the output stream. Our framework only uses this parameter synthesis algorithm as interpolation function on local motion segment. In [15, 29], the state sequence is estimated directly by the phonemes stream, while, in our framework, the state (component) sequence is estimated by a rough lip shape stream, which is inferred by $lip\ shape\ GMMs$. The combination of $lip\ shape\ GMMs$ and $HMMs\ interpolation$ allows us to refine the synthesized motion.

The experiment results on laughter lip animation can be seen in Table 2. 49 laughter episodes are used in total. The results show that our statistical framework outperforms the existing laughter lip animation synthesis based on linear regression model [35]. These two approaches are based on the same dataset and they are capable of synthesizing laughter lip animation in real time. In [35], co-articulation is not taken into account. This may explain why the quality of laughter lip animation is lower with this previous model.

Comparing the results of laughter lip (see Table 2) and speech lip (see Table 1), the quality of laughter lip is not as accurately reproduced as the speech lip. Notice that human speech lip is recorded by motion capture technology while human laughter lip is detected by tracking software. The tracking software cannot precisely capture human lip data. This can have an effect of the the quality of the laughter animation. In our work, we use a unified framework for speech and laughter to model the lip co-articulation. However, lip co-articulation in laughter is much lower than in speech, which is not taken into account in the unified framework. This may explain the difference in error rates of lip shapes motion during speech and laughter.

4. Conclusions

Our work focuses on text-driven lip animation for interactive virtual agents. It involves speech and laughter lip animation synthesis. The unified synthesis framework for speech and laughter lip animation should facilitate the lip animation synthesis for speaking while laughing which we tackle in future work.

The proposed framework consists of $lip\ shape\ GMMs$ and $HMMs\ interpolation$: one is used to infer lip shape from sequence of phonemes; the other to model the co-articulation of human lip motion. The combination of $lip\ shape\ GMMs$ and $HMMs\ interpolation$ ensures to refine the synthesized motion.

The objective evaluation shows that our approach outperforms classical HMM-based synthesis approaches for speech lip animation synthesis and linear regression model for laughter lip animation synthesis.

5. Acknowledgment

This work was partially performed within the Labex SMART (ANR-11-LABX-65) and by the H2020 project ARIA-VALUSPA. The Labex SMART is supported by French state funds managed by the ANR within the Investissements d'Avenir programme under reference ANR-11-IDEX-0004-02.

6. References

[1] H. McGurk and J. W. MacDonald, "Hearing lips and seeing voices," *Nature*, vol. 264, no. 246-248, 1976.

[2] E. Cosatto, J. Ostermann, H. Graf, and J. Schroeter, "Lifelike talking faces for interactive services," *Proceedings of the IEEE, Special Issue on HCMI*, vol. 91, no. 9, pp. 1406–1429, 2003.

[3] E. Cosatto and P. M. Kunt, "Sample-based talking-head synthesis," in PhD Thesis, Signal Processing Lab, Swiss Federal Institute of Techology, Tech. Rep., 2002.

[4] E. Mendi and C. Bayrak, "Facial animation framework for web and mobile platforms," in *e-Health Networking Applications and Services (Healthcom)*, 2011, pp. 52–55.

[5] J. Ostermann and A. Weissenfeld, "Talking faces-technologies and applications," in *ICPR.*, vol. 3, 2004, pp. 826–833.

[6] N. F. Dixon and L. Spitz, "The detection of audiovisual desynchrony," *Perception*, vol. 9, no. 246-248, 1980.

[7] M. M. Cohen and D. W. Massaro, "Modeling coarticulation in synthetic visual speech," in *Models and Techniques in Computer Animation*. Springer-Verlag, 1993, pp. 139–156.

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

[9] S. Fagel, B.-J. Theobald, and G. Bailly, "LIPS 2009: Visual Speech Synthesis Challenge," in *AVSP*, 2009.

[10] E. Bevacqua and C. Pelachaud, "Expressive audio-visual speech," *Journal of Visualization and Computer Animation*, vol. 15, no. 3-4, pp. 297–304, 2004.

[11] E. Bevacqua, M. Mancini, and C. Pelachaud, "Speaking with emotions," in *In Proceedings of the AISB Symposium on Motion, Emotion and Cognition*, 2004, pp. 197–214.

[12] Z. Deng, U. Neumann, J. P. Lewis, T. Kim, M. Bulut, and S. Narayanan, "Expressive facial animation synthesis by learning speech coarticulation and expression spaces," *IEEE Trans. Vis. Comput. Graph.*, vol. 12, no. 6, pp. 1523–1534, 2006.

[13] C. Luo, J. Yu, and Z. Wang, "Synthesizing real-time speech-driven facial animation," in *ICASSP*, 2014, pp. 4568–4572.

[14] G. Hofer and K. Richmond, "Comparison of hmm and tmdn methods for lip synchronisation," in *in Proc. Interspeech*, 2010, pp. 454–457.

[15] G. Hofer, J. Yamagishi, and H. Shimodaira, "Speech-driven lip motion generation with a trajectory hmm," in *in Proc. Interspeech*, 2008, pp. 2314–2317.

[16] M. Brand, "Voice puppetry," in *SIGGRAPH*, 1999, pp. 21–28.

[17] Y. Li and H. yeung Shum, "Learning dynamic audio-visual mapping with input/output hidden markov models," *IEEE Trans. on Multimedia*, pp. 542–549, 2006.

[18] T. Ezzat, G. Geiger, and T. Poggio, "Trainable videorealistic speech animation," in *ACM Trans. Graph.*, 2002, pp. 388–398.

[19] J. Jia, S. Zhang, F. Meng, Y. Wang, and L. Cai, "Emotional audio-visual speech synthesis based on pad," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 3, pp. 570–582, 2011.

[20] L. Wang, W. Han, X. Qian, and F. Soong, "Synthesizing photo-real talking head via trajectory-guided sample selection," in *Interspeech*, 2010, pp. 446–449.

[21] L. Wang and F. K. Soong, "Hmm trajectory-guided sample selection for photo-realistic talking head," *Multimedia Tools and Applications*, May 2014.

[22] X. Zhang, L. Wang, G. Li, F. Seide, and F. K. Soong, "A new language independent, photo-realistic talking head driven by voice only," in *Interspeech2013*, 2013, pp. 2743–2747.

[23] L. Wang, X. Qian, W. Han, and F. K. Soong, "Photo-real lips synthesis with trajectory-guided sample selection," in *The Seventh ISCA Tutorial and Research Workshop on Speech Synthesis, Kyoto, Japan, September 22-24, 2010*, 2010, pp. 217–222.

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

[25] L. Wang, W. Han, F. K. Soong, and Q. Huo, "Text driven 3d photo-realistic talking head," in *Interspeech*, 2011, pp. 3307–3308.

[26] L. Wang, X. Qian, L. Ma, Y. Qian, Y. Chen, and F. K. Soong, "A real-time text to audio-visual speech synthesis system," in *Interspeech2008*, 2008, pp. 2338–2341.

[27] W. Han, L. Wang, F. K. Soong, and B. Yuan, "Improved minimum converted trajectory error training for real-time speech-to-lips conversion," in *ICASSP*, 2012, pp. 4513–4516.

[28] Z.-H. Ling, K. Richmond, J. Yamagishi, and R.-H. Wang, "Articulatory control of hmm-based parametric speech synthesis driven by phonetic knowledge," in *INTERSPEECH*, 2008, pp. 573–576.

[29] L. Wang, H. Chen, S. Li, and H. M. Meng, "Phoneme-level articulatory animation in pronunciation training," *Speech Communication*, vol. 54, no. 7, pp. 845 – 856, 2012.

[30] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for hmm-based speech synthesis," in *ICASSP*, vol. 3, 2000, pp. 1315–1318 vol.3.

[31] R. Kent and F. Minifie, "Coarticulation in recent speech production models," in *Journal of Phonetics*, vol. 5, 1977, pp. 115–135.

[32] G. Bailly, A. Bgault, F. Elisei, and P. Badin, "Speaking with smile or disgust: data and models," in *AVSP'08*, 2008, pp. 111–114.

[33] D. Cosker and J. Edge, "Laughing, crying, sneezing and yawning: Automatic voice driven animation of non-speech articulations," *CASA*, pp. 21–24, 2009.

[34] H. Çakmak, J. Urbain, J. Tilmanne, and T. Dutoit, "Evaluation of hmm-based visual laughter synthesis," in *ICASSP*, 2014, pp. 4578–4582.

[35] Y. Ding, K. Prepin, J. Huang, C. Pelachaud, and T. Artières, "Laughter animation synthesis," in *AAMAS*, 2014, pp. 773–780.

[36] C. Busso, M. Bulut, C.-C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. N. Chang, S. Lee, and S. S. Narayanan, "Iemocap: Interactive emotional dyadic motion capture database," *Language resources and evaluation*, vol. 42, no. 4, pp. 335–359, 2008.

[37] I. Pandzic and R. Forcheimer, *MPEG4 Facial Animation - The standard, implementations and applications*. John Wiley & Sons, 2002.

[38] G. Bailly, O. Govokhina, F. Elisei, and G. Breton, "Lip-synching using speaker-specific articulation, shape and appearance models," *EURASIP J. Audio, Speech and Music Processing*, vol. 2009, 2009.

[39] K. Sum, W. Lau, S. Leung, A. Liew, and K. Tse, "A new optimization procedure for extracting the point-based lip contour using active shape model," in *ICASSP*, vol. 3, 2001, pp. 1485–1488 vol.3.

[40] "Ubiqus: 2008," *http://www.ubiqus.com/.*, Retrieved September 11th 2008.

[41] J. Urbain, R. Niewiadomski, E. Bevacqua, T. Dutoit, A. Moinet, C. Pelachaud, B. Picart, J. Tilmanne, and J. Wagner, "Avlaughtercycle," vol. 4, no. 1. Journal on Multimodal User Interfaces, 2010, pp. 47–58.

[42] J. Saragih, S. Lucey, and J. Cohn, "Deformable model fitting by regularized landmark mean-shift," *International Journal of Computer Vision*, vol. 91, no. 2, pp. 200–215, 2011.

[43] J. Urbain, H. Çakmak, and T. Dutoit, "Automatic phonetic transcription of laughter and its application to laughter synthesis," in *ACII*, 2013, pp. 153–158.

[44] Y. Linde, A. Buzo, and R. Gray, "An algorithm for vector quantizer design," *Communications, IEEE Transactions on*, vol. 28, no. 1, pp. 84–95, Jan 1980.

[45] R. H. Stetson, in *Motor Phonetics: A Study of Speech Movements in Action*. North Holland Publishing Company, 1951.