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 depends on the context of the surrounding phonemes [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 of neighboring phonemes are not explicitly specified. Such 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, 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 phoneme “s” 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
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 Ubiqu [40]. In total, 49 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 (HMMs 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. Figure 3 illustrates the outputs of lip shape GMMs and HMMs interpolation. The framework overview is illustrated in Figure 1.

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
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 a viseme is defined by the temporal value of the maximum of the phoneme segment. We consider the maximum point on the envelop. The apex of a viseme is defined by the temporal value

2.1.1. Training: Building GMMs

Each phoneme is also characterized by its duration, \( d \). For each phoneme label, we have a set of \( m_c \) and \( d_c \). This set is noted by \( m_d \).

\[ P(m_d, \lambda | m_d) = \text{const} \]  
(1)

where \( m_d \) is the transpose of the vector. The parameter set of the GMM is \( \lambda \). 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 \( \alpha_c \). The distribution with mean \( \mu_c \) and covariance \( \sigma_c \) is denoted as \( N(\mu_c, \sigma_c) \).

Given the input of phonemes sequence, we can align a sequence of \( GMM_{ d} \) according to phoneme label. Then each \( GMM_{ d} \) is used to synthesize the segment of lip animation. The following introduces how to synthesize lip animation from a given \( GMM_{ d} \).

Given the phoneme duration, \( d \), \( GMM_{ d} \) is used to select the most likely component as follows:

\[ c = \arg\max_{\alpha_c} P(d|\mu_c, \sigma_c) \]  
(2)

where the \( c \)-th component is determined as the occurring component. Then the occurring component is applied to \( GMM_{ d} \).

Once the occurring component is determined, the mean vector, \( \mu_c \), of the \( c \)-th component in \( GMM_{ d} \) 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. 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, \( \Delta m^l \), 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 $HMM_{s}$ interpolation can be built separately for 13-dimensional motion.

2.2.2. Synthesis: Smoothing the lip shape stream

During the synthesis step, one $HMM_{ml}$ with observation, $m_l$, can be extracted from the trained $HMM_{ml}$ with observation, $o_l$, by ignoring the observations $\Delta m_l$ and $\Delta o_l$.

As introduced in 2.1, lip shape $GMM$s 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 $HMM_{ml}$ by using Viterbi algorithm. $q$ is applied to $GMM_{ml}$. Given $GMM_{ml}$, and a state sequence, $q$, a smooth trajectory, $m_l$, can be synthesized directly by solving the equation as follows:

$$\frac{\partial \log P(W^m|q, HMM_{ml})}{\partial m_l} = 0$$

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 $GMM$s; Section 2.2 explained the HMMs-based interpolation function ($HMM$ interpolation). Generally, once lip shape stream is inferred by lip shape $GMM$s; it can be smoothed by $HMM$ interpolation. The following details how to perform the lip animation synthesis in real time using lip shape $GMM$s and $HMM$s 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-Consonant-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.

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:

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

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 $GMM$s, but also it takes into account co-articulation of close neighboring
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

<table>
<thead>
<tr>
<th></th>
<th>Proposed model</th>
<th>HMM-based framework[29]</th>
</tr>
</thead>
<tbody>
<tr>
<td>lip openness</td>
<td>0.049 (0.029)</td>
<td>0.19 (0.056)</td>
</tr>
<tr>
<td>lip extension</td>
<td>0.061 (0.041)</td>
<td>0.23 (0.064)</td>
</tr>
<tr>
<td>top lip protrusion</td>
<td>0.163 (0.055)</td>
<td>0.34 (0.071)</td>
</tr>
<tr>
<td>bottom lip protrusion</td>
<td>0.132 (0.047)</td>
<td>0.31 (0.031)</td>
</tr>
</tbody>
</table>

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).

<table>
<thead>
<tr>
<th></th>
<th>Proposed model</th>
<th>regression model[35]</th>
</tr>
</thead>
<tbody>
<tr>
<td>lip openness</td>
<td>0.18 (0.076)</td>
<td>0.22 (0.067)</td>
</tr>
<tr>
<td>lip extension</td>
<td>0.19 (0.039)</td>
<td>0.25 (0.059)</td>
</tr>
<tr>
<td>top lip protrusion</td>
<td>0.21 (0.061)</td>
<td>0.26 (0.024)</td>
</tr>
<tr>
<td>bottom lip protrusion</td>
<td>0.27 (0.059)</td>
<td>0.31 (0.041)</td>
</tr>
</tbody>
</table>

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


