Web-based live speech-driven lip-sync

An audio-driven rule-based approach

Gerard Llorach*, Alun Evans, Josep Blat
Interactive Technololgies Group, Universitat Pompeu Fabra
Barcelona, Spain
{gerard.llorach, alun.evans, josep.blat}@upf.edu

Giso Grimm, Volker Hohmann
*Medical Physics,Carl von Ossietzky-Universität Oldenburg
Oldenburg, Germany
{g.grimm, volker.hohmann}@uni-oldenburg.de

***Abstract*— Virtual characters are an integral part of many games and virtual worlds. The ability to accurately synchronize lip movement to audio speech is an important aspect in the believability of the character. In this paper we propose a simple rule-based lip-syncing algorithm for virtual agents using the web browser. It works in real-time with live input, unlike most current lip-syncing proposals, which may require considerable amounts of computation, expertise and time to set up. Our method generates reliable speech animation based on live speech using three blend shapes and no training, and it only needs manual adjustment of three parameters for each speaker (sensitivity, smoothness and vocal tract length). Our proposal is based on the limited real-time audio processing functions of the client web browser (thus, the algorithm needs to be simple), but this facilitates the use of web based embodied conversational agents.**

Keywords*— virtual characters, lip synchronization, visual speech synthesis

I. INTRODUCTION

In recent years, virtual characters have been increasingly appearing in different applications which require the reproduction of speech content in real-time. It is well established that non-verbal aspects play an important role in communication [1]. Thus, to enhance believability, virtual characters need to move their lips according to what they are saying, among other features such as suitable gestures and facial expressions. Using virtual characters instead of speech-only communication can improve the human-computer interaction in several aspects [2] and many virtual world applications can benefit from lip-syncing: video phone calls via avatars, tutors for hearing-impaired, embodied conversational agents and even movie and computer games virtual characters.

Believable lip movements according to speech have an important role as speech is perceived through both visual and audio cues together: the McGurk effect [3] proves that the combination of discrepant acoustic and visual cues can change the perception of the acoustic signal. Additionally, accurate lip syncing for virtual humans can help the hearing-impaired and improve intelligibility in noisy situations [4] [5].

Lip-syncing has been an active research area for many years, where knowledge and expertise in speech processing, computer graphics and even psychology are required. We review below several lip-sync techniques based on different approaches that have been proposed in recent years. Although many solutions provide real-time lip-sync, usually their parameters are generated using prerecorded speech and then the lip-sync is reproduced in real-time. Few solutions generate lip-sync based on live input, and due to the complexity of their algorithms, they cannot work in the web browser.

In light of the importance of using virtual characters for human-computer interaction and virtual worlds our work focuses on developing virtual characters in the most common internet application, the web browser. Web browser technologies have been improving in the last years: the introduction of WebGL made possible the rendering of complex scenes [6] and the current Web Audio API [7] provides several real-time audio processing functions implemented in assembly/C/C++ code.

Currently, there are no applications capable of real-time lip-sync using the web browser. This paper introduces a novel web-based and real time lip-sync technique, suitable for use with web-based virtual characters. It is based on computing the Fourier transform of the audio input to obtain the energies of different frequency bands, which are then mapped through a set of rules to three visual parameters, the weights of three blend shapes. Because our system is speaker dependent, the user can modify three parameters to improve the lip-sync: sensitivity threshold, smoothness and vocal tract length. Our solution generates and reproduces lip-sync with no perceptible delay. However, due to the processing limitations of the web context, certain lip configurations are not possible.

II. RELATED WORK

A. Acoustic features / phonemes extraction

Over the last decades several methods have been proposed, with two established approaches to analyze speech. In one of them, acoustic features such as fundamental frequency, Linear Predictive Coding (LPC) and Mel-Frequency Cepstral Coefficients (MFCCs), among others, are extracted from the speech signal [8]. When using audio-driven articulation, additional acoustic features can be extracted, such as prosody and intensity, which can change the lip-sync and overall facial expression [9]. However, these methods may be highly speaker dependent as they rely on the training data [10] and can be difficult to implement as they may require extensive expertise.

In the other approach, custom or external phoneme recognition algorithms are used. Due to the recent advances in automatic speech recognition [20], mature algorithms can be used to extract phonemes. Phoneme-driven articulation has the advantage of being speaker independent in comparison with audio-driven articulation, but it does not take into account

prosodic acoustic information and it cannot be used in real-time, as the next phoneme is not available until recognized.

B. Visual features mapping

Once the speech has been analyzed, the extracted audio features or phonemes are mapped to lip parameters of the virtual character, using visemes, action units, codewords, control parameters, etc. These visual parameters define the deformations of the face and can have different levels of complexity. For example, some researchers use the standard MPEG-4 [12], which defines about 66 displacements and rotations of feature points of the face, while others use a small set of predefined facial deformations or blend shapes [13].

When using multiple visual parameters, manually mapping audio-features or phonemes to these parameters might be very cumbersome. Therefore, data-driven audio-visual mappings have been used in most recent research approaches, both audio-driven and phoneme-driven articulations [14, 10]. Data-driven techniques use recorded corpora to extract visual parameters which are associated to the acoustic features or phonemes. Results can achieve high accuracy, but as the techniques rely on recorded corpora they can create speaker dependent animations, as lip trajectories contain information about a person's identity [15]. Thus, transferring these visual parameters to characters with different visual features can be complex, even more when using non-human characters.

Most approaches that use manual mappings associate phonemes to specific blend shapes or predefined visemes, so that they can deal with a comprehensive small set of acoustic and visual parameters. These approaches do not need to analyze a recorded corpus, which can reduce the amount of necessary work and material. For example, [16] requires manually designing 441 animations between phonemes, and [13] uses 16 predefined blend shapes to map phonemes.

C. Real-time lip-sync

Although several solutions dealing with live input speech have been proposed, few have received attention. One of the few real-time phoneme-driven approaches [11] uses a prediction table to try to estimate the next facial deformation without knowing the forthcoming phoneme. Almost all the other solutions dealing with live input are audio-driven, as the recognition algorithm introduces no visual delay and several audio frames can be computed and blended for each video frame [14, 17]. Nevertheless, in human communication, visual speech comes before audio speech [18], and it can be argued that it is nearly impossible to perfectly match and generate lip-sync in real-time with live input, as visual speech should start a few milliseconds before the audio speech. However, [14] proved that real-time lip-sync is still a valid solution, as their real-time lip-sync improved speech intelligibility.

III. Method Overview

Our approach is based on manually mapping simple audio features to very few visual parameters, namely, the weights of three blend shapes. Most audio-driven articulations extract complex audio features which cannot be manually mapped. Moreover, speech recognition algorithms use frequency features, such as MFCCs and LPC, which cannot be extracted in real-time through the web browser API. We use the real-time available data, the short-term spectrum, and compute the energies of several frequency bands to drive the blend shapes.

Our mapping aims to relate different frequency-band specific energies to visual parameters. Focusing initially on the vowels (and later on the consonants), we define a series of frequency-based rules that drive the weights of three blend shapes corresponding to different lip configurations: *kiss*, *mouth open* and *lips pressed*. They control respectively the horizontal aperture of the mouth, its vertical aperture and the volume of the lips, and thus a variety of lip configurations can be achieved.

The user can change three acoustic parameters: vocal tract length (speaker dependent), smoothness (language/speaker dependent) and sensitivity threshold (signal dependent). As the speech frequencies and features can be different for each person [19], we permit to scale them according to a vocal tract length factor. We also permit to change the smoothness of the animation, as speakers with a higher speech rate might need a more reactive lip-sync. We want our system to work with different microphones and situations, so a sensitivity threshold can be adjusted to avoid undesired background noise.

IV. Audio Processing

In this paper we use a simple energy-based vocal tract model. We estimate the energy of the formants to produce visual features. We extract a smoothed short-term power spectrum density (st-PSD) in real-time by means of a function provided by the Web Audio API [7]. This closed function processes each audio block through the following steps: the input samples are windowed with a Blackman window and then the Fast Fourier Transform (FFT) is computed. The output st-PSD is smoothed over time with previous outputs (1) and then converted to dB (2). We used audio blocks of 1024 samples at a sampling rate of 44.1 kHz.

A. Web Audio API FFT

We do not detail Blackman windowing or FFT as these steps are well known, and not customizable. Smoothing over time involves using the previously smoothed data:

$$\hat{X}[k] = \tau\, \hat{X}_{-1}[k] + (1-\tau)|X[k]|, \;\; for \begin{cases} k = 0, \dots, N-1 \\ 0 < \tau < 1 \end{cases} \qquad (1)$$

where $X[k]$ is the complex frequency domain data computed, $\hat{X}[k]$ is the smoothed one and τ is a smoothing variable defined by the user, which ranges between 0 and 1. The final step is the dB conversion, which provides $Y[k]$, the output of the Web Audio API function:

$$Y[k] = 20\, \log_{10} \hat{X}[k], \quad for\ k = 0, \dots, N-1 \qquad (2)$$

B. Bounding frequencies and energy bins

The output of this function, which ranges between -25dB and -160dB approximately for properly scaled speech signals and the applied FFT length, is scaled and increased by a user defined sensitivity threshold δ (set to 0.5 by default) through (3) to map the dynamic range to the interval [-0.5,0.5].

$$\dot{Y}[k] = \delta + (Y[k] + 20)/140, \quad for\ k = 0, \dots, N-1 \qquad (3)$$

giving the processed smoothed st-PSD $\dot{Y}[k]$.

The processed st-PSD (3) is divided into frequency bands, and the energy is computed for each one. We used log-scaled data instead of signal intensity to compute the energy because it is the available output of the API function. These bands have been defined empirically (see below), and the bounding frequencies can be scaled with a user defined vocal tract length factor γ. The bounding frequencies F_{bins} are defined by (4):

$$F_{bins} = [0, 500\gamma, 700\gamma, 3000\gamma, 6000\gamma]\ Hz \tag{4}$$

In our experiments we used the length factors 1 for females and 0.8 for males. These bounding frequencies are then transformed to frequency data indices F_{ind} :

$$F_{ind}[m] = \frac{2N}{fs} F_{bins}[m], \quad for\ m = 0, \dots, M-1 \tag{5}$$

where fs is the sampling frequency, M is the number of bounding frequencies and N is the number of samples per audio block. The energy for each bin $E[m]$ is computed only taking into account positive values of the processed st-PSD:

$$E[m] = \frac{1}{F_{ind}[m+1] - F_{ind}[m]} \sum_{j=F_{ind}[m]}^{F_{ind}[m+1]} \dot{Y}[j], \quad for \begin{cases} \dot{Y}[j] > 0 \\ m = 0, \dots, M-2 \end{cases} \tag{6}$$

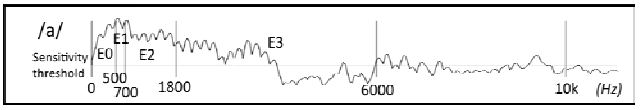

Fig. 1. Example of the processed st-PSD (3) for phoneme /a/, with divisions in different energy bins by bounding frequencies with a factor of 1.

One of the crucial parts of our algorithm is the definition of different energy bins and bounding frequencies. In our system, we focused on the energy and frequency of the formants for each vowel, without taking into account the fundamental frequency component. We used references for average vowel formants (Table 1) and analyzed the energy that each vowel would produce in different frequency regions through observation.

TABLE I. AVERAGE VOWEL FORMANTS [19]

Vowel (IPA)	Formant F1 (Hz)	Formant F2 (Hz)
/a/	850	1610
/e/	390	2300
/i/	240	2400
/o/	360	640
/u/	250	595

It is important to note that the frequencies for each formant will vary depending on the speaker. Males tend to have a longer vocal tract than females and may require a smaller factor. The following paragraph will be explained as if the vocal tract length factor was set to 1 (4).

The first energy bin, between 0 and 500 Hz, represents the fundamental frequency and low-frequency formants. We discarded this first energy bin in the visual mapping, as it is not a distinctive feature. The second energy bin, between 500 and 700 Hz, is quite crucial together with the third bin (700 to 1800 Hz), as both contain most of the formants. We found that these bins would be different across the vowels and could provide important information for the visual mapping. The fourth and last bin, between 1800 and 6000 Hz, mostly represents the second formants of the vowels /e/ and /i/ and partly the energy for some fricatives such as /s/ and /f/.

V. VISUAL MAPPING

The system has to be as 'speaker independent' as possible, and associating specific frequency features to detailed visual features can generate wrong animations when the system is used by different speakers. Thus, we kept our visual features as simple as possible, using just three blend shapes. Opening the mouth is the most basic one, all the vowels use this blend shape. The kissing blend shape is used to differentiate between vowels, /o/ and /u/ are representative of this blend shape. For fricative consonants, the lips pressed blend shape is used. We created a set of rules to drive these blend shapes according to the energy of the frequency areas.

Fig. 2. From left to right: default face, kiss blend shape, lips pressed blend shape, mouth open blend shape.

Once the energy from the frequency bands has been computed, the three visual parameters are computed using the equations:

$$BS_{kiss} = \begin{cases} 1 - 2E[2], & for\ E[1] \geq 0.2 \\ (1 - 2E[2]) \cdot 5E[1], & for\ E[1] < 0.2 \end{cases} \tag{7}$$

$$BS_{lips} = 3E[3] \tag{8}$$

$$BS_{mouth} = 0.8(E[1] - E[3]) \tag{9}$$

where E[i] is the energy for bin i, BS_{kiss}, BS_{lips} and BS_{mouth} are the respective weights for the kiss, lips closed and mouth open blend shapes.

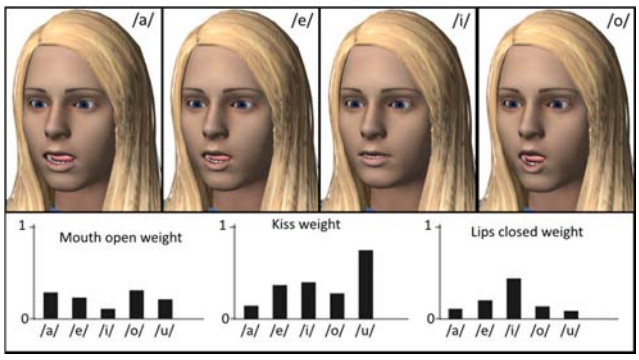

Fig. 3. Example of blend shape weights for different vowels.

VI. Implementation

We implemented a web-based system using a 3D scene editor and engine, WebGLStudio [21], which supports blend shapes and animations, necessary for virtual characters and real-time lip-sync. We used the current Web Audio API [7] to process the audio in real time.

In addition, we also implemented the system in a non-web-based system. We used the game engine from Blender [22] to render the characters and apply the blend shapes in real-time. We implemented the real-time audio processing within a real-time 3D audio rendering engine, TASCAR [23]. Since clean speech signals are available in virtual acoustic environments, the sensitivity to noise is not relevant in this context.

VII. Results and Discussion

We tested both systems with live input and different speakers and with several speech audio files. The quality of computer generated characters has increased enormously over the years, thus making it difficult to compare the results of subjective quality ratings of virtual characters. In consequence, using subjective quality scores is not the best evaluation, as these scores would change over the years and would depend on the appearance of the virtual character. In order to provide an objective evaluation we are preparing a speech intelligibility test, as in [14].

Compared with other lip-sync systems, the solution that we present is fast and simple to implement. A high level of expertise is not required and there is no need to use a corpus to train the system. The lip-sync can be applied to several different characters with little effort. From the implementation point of view, there are many libraries that implement the FFT in real-time and our frequency rules are quite straightforward to implement. In order to achieve better performance, the user can adjust the three acoustic parameters (sensitivity threshold, smoothness and vocal tract length) to improve the lip-sync result. Our solution is then very useful for many applications that do not require high accuracy lip-sync and cannot spend many resources in it.

Many proposed lip-sync solutions focused only in the audio-to-visual mapping. The configuration of the lips is highly affected by prosody and mood, therefore reproducing lip-sync without taking into account prosody or facial expressions would be a quite limited approach, similar to generating speech without any prosody. Thus, it is important that the lip-sync is easy to integrate with existing facial animation systems. Our approach can be easily integrated with facial animation systems that use blend shapes, as facial blend shapes can be applied at the same time to change the overall facial expression or slightly modify the lip-sync.

Acknowledgments

This research has been partially funded by the Spanish Ministry of Economy and Competitiveness (RESET TIN2014-53199-C3-3-R), by the DFG research grant FOR173 and by the European Commission under the contract number H2020-645012-RIA (KRISTINA).

References

[1] M. A. Hechtand and N. Ambady, "Nonverbal communication and psychology: Past and future," *The New Jersey Journal of Communication*, **7**(2), 1-12, 1999.

[2] J. Ostermann and A. Weissenfeld, "Talking faces technologies and applications," *Proceedings of the International Conference on Pattern Recognition*, Cambridge, UK, vol 3, pp. 826– 833, 2004

[3] H. McGurk and J. MacDonald, "Hearing lips and seeing voices," *Nature*, **264** , pp. 746–748, 1976.

[4] A. MacLeod and Q. Summerfield. "Quantifying the contribution of vision to speech perception in noise." *Br. J. Audiol.* **21**, pp. 131–141. 1987.

[5] D. W. Massaro and J. A. Simpson, *Speech Perception by Ear and Eye: A Paradigm for Psychological Inquiry*, Hillsdale, NJ: Lawrence Erlbaum Associates, 1987.

[6] A. Evans, M. Romeo, A. Bahremand, J. Agenjo, J. Blat, "3D Graphics on the Web: A Survey," *Computers & Graphics* **41**, pp 43-61. 2014.

[7] World Wide Web Consortium (W3C). "Web Audio API," <http://webaudio.github.io/web-audio-api/>, 2016.

[8] P. Kakumanu, A. Esposito, O. N. Garcia, R. Gutierrez-Osuna, "A comparison of acoustic coding models for speech-driven facial animation," *Speech Communication* **48**(6), pp. 598-615, 2006.

[9] R. Gutierrez-Osuna et al., "Speech-Driven Facial Animation with Realistic Dynamics," *IEEE Trans. Multimedia* **7**(1), pp. 33–42, 2005.

[10] S.L. Taylor, "Dynamic Units of Visual Speech," in *Proc. 11th ACM SIGGRAPH/Eurographics Conf. Computer Animation,* pp. 275–284, 2012.

[11] L. Wei and Z. Deng, "A Practical Model for Live Speech-Driven Lip-Sync," in *IEEE Computer Graphics and Applications* **35**(2), pp. 70-78, Mar.-Apr. 2015.

[12] A.M. Tekalp and J. Ostermann, "Face and 2-D mesh animation in MPEG-4", *Signal Processing: Image Comm.* **15**, pp. 387–421, 2000.

[13] A. Wang, M. Emmi and P. Faloutsos, "Assembling an expressive facial animation system," in *Proceedings of the 2007 ACM SIGGRAPH symposium on Video games (Sandbox '07)*. ACM, New York, NY, USA, pp. 21-26, 2007.

[14] J. Liu, M. You, C. Chen, M. Song, "Real-time speech-driven animation of expressive talking faces," *International Journal of General Systems* **40**, pp. 439-455, 2011.

[15] P. Jourlin, J. Luettin, D. Genoud and H. Wassner, "Acoustic-labial speaker verification," *Pattern Recognition Letters* **18**, pp. 853–858, 1997.

[16] Y. Xu, A. W. Feng, S. Marsella and A. Shapiro, "A Practical and Configurable Lip Sync Method for Games," In *Proceedings of Motion on Games (MIG '13),* pp. 131-140, 2013.

[17] P. Hong, Z. Wen, and T.S. Huang, "Real-Time Speech-Driven Face Animation with Expressions Using Neural Networks," *IEEE Trans. Neural Networks*, **13**(4), pp. 916–927, 2002.

[18] J. H. Venezia, S. M. Thurman, W. Matchin, S. E. George and G. Hickok, "Timing in audiovisual speech perception: A mini review and new psychophysical data," *Atten Percept Psychophys* **78**(2), pp. 583-601, 2016.

[19] J.C. Catford, *A Practical Introduction to Phonetics*, Oxford University Press, p. 161, 1988.

[20] F. Mihelic and J. Zibert, *Speech Recognition*, InTech, 2008.

[21] J. Agenjo, A. Evans and J. Blat, "WebGLStudio: a pipeline for WebGL scene creation," Proceedings of the 18th International Conference on 3D Web Technology. ACM, 2013.

[22] T. Roosendaal, "Blender," Blender Foundation <https://www.blender.org/>, 1995.

[23] G. Grimm, J. Luberadzka, T. Herzke and V. Hohmann, "Toolbox for acoustic scene creation and rendering (TASCAR) - Render methods and research applications," in Proceedings of the Linux Audio Conference, Mainz, 2015.