Automatic Visual Speech Animation

José Serra\textsuperscript{1}, João Freitas\textsuperscript{2,3}, Miguel Dias\textsuperscript{2,3}, and Verónica Orvalho\textsuperscript{1}

\textsuperscript{1}Instituto de Telecomunicações
\textsuperscript{2}Microsoft Language Development Center, Tagus Park, Porto Salvo, Portugal
\textsuperscript{3}ISCTE-Lisbon University Institute/ADETTI-IUL, Lisboa, Portugal
jserra@dcc.fc.up.pt, i-joaof@microsoft.com, miguel.dias@microsoft.com, veronica.orvalho@dcc.fc.up.pt

Abstract. Visual speech animation, also known as, lip synchronization is the process of matching a speech audio file with the lips' movements of a synthetic character. Visual speech is a very demanding task, being done either fully manual, which is very time consuming, or with automatic methods based on data analysis. Currently there is still no automatic method that generates any sequence of visual speech without requiring further fine tuning. This research focused on the problem of automatically achieving lip-sync and led to a system that relies on speech recognition to obtain the words from audio and maps them to the visual poses, thus automatically obtaining visual speech animation. The system also supports language translation, which allows animation in a language different from the one spoken. Automatic visual speech animation has great impact in the entertainment industry, where it can reduce the time required to produce the animation of talking characters.

Keywords: Computer animation, Automatic Speech Animation, Speech Translation, Speech Recognition

1 Introduction

Speech is a key element to convey the ideas and thoughts of the person behind any 3D talking character. However, the speech is not composed only by sounds, but also by their poses, which in fact, contribute decisively to the believability of the virtual characters. If the speech animation is not done correctly, i.e. the lips movements do not resemble the ones of a person or if these are not synchronized with the audio, the viewers will find the animation awkward.

A speech sequence can be discretized as a sequence of sounds, also known as phones, and silences. Each phone has associated facial poses, which are positions of the visible articulators of the vocal track, being composed by lips, teeth and tongue. All vocal track articulator can influence the production of a phone, however not all are visible, therefore different phones have the same pose. Visemes are a key concept in visual speech animation. It is important to note that it is not possible to directly generate visual speech by concatenating the visemes, as the produced animation will be over-articulated. This results from the co-articulation, which refers to the effect one phone, and its corresponding viseme,
has over the ones around it. Therefore, speech animation has two major components, the synchronization of the audio-lips movements and the co-articulation simulation, which a digital artist has to take into account when animating a speech. Traditionally, an artist has to decompose the audio in its phones, reduce them, as not all influence visual speech, mark them in the sound wave, create the time-line with the visemes and finally control the influence of each in the animation [1]. This is an extremely slow and time consuming process, which takes about 25 to 30 minutes to animate a sentence.

Several techniques have emerged that try to automate different stages of the speech animation pipeline, such as lips-audio synchronization, also known as data-driven techniques. These have the goal of reducing the speech animation costs. It is within data driven techniques this research made its contribution, with the goal of creating a fully automatic visual speech animation system, capable of generating the 3D animation data from a speech audio signal. Language translation was also added to the equation, as a consequence of the need in the videogame and film industry to animate their products in different languages. This is a problem because each time, e.g. a film, is translated all the animation has to be done from scratch for each language, as it usually is not possible to reuse it in different languages. Thus, automatic speech animation and translation largely helps to reduce the time required to animate a film or video-game.

The proposed system is based on automatic speech recognition, which is used to obtain the words both in Portuguese and in English. The audio is analysed to get the orthographic and phonetic transcriptions, the text is then translated if necessary also generating the new audio and the visemes that map to the found phones are chosen. With the system it is possible to automatically generate visual speech. The following sections are organized as it follows: state of the art with the description of what has been done in speech animation, system description, which contains a more detailed description of how the system was implemented, results & discussion, with the main results obtained, how they were obtained and their analysis and finally the conclusion and future work.

## 2 State of the art

Automatic speech animation, in contrast to traditional animation, does not rely on direct work by the animator. It is based on different sources of information, which are then mapped to a model and processed to generate the animation. Based on the sources it is possible to divide data-driven speech animation in two types: performance-based or MoCap and speech based. Performance based animation tracks the movements of a person and maps them to face model, using video or infra-red signal analysis. The tracking can be based on marker placed on the actors [4], or markerless, such as [6]. However, to be able to correctly track the actor, special equipment and very controlled environments are required, thus having high costs. Using Mocap all the facial movements of the actor are captured, therefore the lips movements and audio synchronization are automatically achieved. Fine tuning is still required as the tongue and inner lips movements are
not captured. On the other hand, speech based animation relies on the analysis of text and/or audio to generate the animation data. One of the main advantages, compared to MoCap, is the reduction of special equipment needed, as only a microphone is required. The present work falls within the second category.

Visual speech animation research is divided in two major areas, depending on the mapping between the input and the visual poses, being them: phone-to-viseme map and sub-phonetic map. In the first, the phones are obtained from text or audio analysis, mapped to the visemes and finally assembled in a timeline. However, the speech animation process is composed by two parts: lip-sync and coarticulation and with this example the co-articulation is not simulated, therefore the animation will look over-articulated. One common approach is to use triphones [2, 3] instead of only phones. As the triphones contain all the motion information, the coarticulation is automatically modelled. The major problem with triphones is the creation of the database that is very costly, e.g. with a 40 phone language there will be roughly $40^3$ triphones. Another approach is to use a model to simulate this effect and one of the most common is the Cohen and Massaro [5] model, where each phone has a curve of influence that can be larger than the duration of the phone, thus simulating the co-articulation effect. The major problem with this model is that it is language dependent. It is important to note that to date there is still not any model capable of simulating all the possible effects. Regarding the lip-sync, most approaches today use a text-to-speech (TTS) systems that take a text as input and generates the audio, with the goal of getting the phones and their durations, e.g. in [7]. However, with a TTS, it is not possible to create a fully automatic speech animation system as the text has to be manually inserted and the audio will have a slightly robotic tone. Another approach is to use automatic speech recognition (ASR) to get the phones and their timings, which solves this problem. Unfortunately ASR is still far from perfect, at least for general dictation, however if the context is well defined the results of the recognition tend to be very good.

Sub-phonetic mapping tries to simulate the behaviour of continuous speech. By directly mapping the sub-phonetic features to the visemes, more detail can be obtained from a sound wave, as the in-between of the phones can also be used to generate data for the animation, thus automatically simulating the coarticulation. Another advantage is the increase of performance, as no phonetic discretization is required. The major problem with sub-phonetic methods is the high sensitivity to noise. Here the map between the sound features and the poses is learned via machine learning, such as in [8] where the sound features were associated to the facial features using PCA and nearest-neighbour mapping. [9] used a set of classifiers to create a real-time speech animation system.

The proposed system uses ASR to get the orthographic transcription of what was spoken. Using the audio as input it is possible to create a fully visual speech animation system. It is based on a previously created phone-to-viseme map that will allow the generation of the animation data from the words recognized.
3 System Description

The created system is divided in two major components: 1) a speech processing tool, responsible for all the input analysis and choice of data that will drive the animation, and 2) a plugin inserted in an animation engine, which has the goal of connecting both. These components lead to a logical division between processing and animation as all the animation data is generated in an entirely independent tool, thus making the system independent of the animation engine. Translation was also added to the created system mainly due to its application in the entertainment industry, allowing instant translation of both the audio and facial animation. It is necessary to note that, when translating a speech, all the input audio is made useless as it can be used in the animation. Therefore, it is necessary to use a text-to-speech to produce the new audio that will drive the animation data. The two components of the system are now described with the whole pipeline in the Fig. 1.

3.1 Speech Processing Tool

The initial behaviour of the tool varies according to the fact that it is translating the input or not. All the system is controlled by a core, responsible for the flow of information between the different modules, which made the system modular enough to allow a person to change the approach of any specific module and keep the system work as intended.

Speech Recognition

The input module was implemented using Microsoft Speech API (SAPI 5.4) [13], more precisely the SR engine available within SAPI. It can be used either in dictation or command and control mode. The first allows the recognition of any utterance, however it has higher error rates, compared to the second, which allows the definition of commands, sentences, being the only ones recognized by the system. Both modes can be used by the system in Portuguese (pt-pt) or English (en-us). However, the design of the pt-pt language model for the dictation scenario, included essentially unigrams bigrams and trigrams of common words of the Portuguese vocabulary, telephone numbers, proper names of people and some business names and addresses of the Portuguese culture and market, and was not yet trained with a general purpose large text corpus. Therefore, the word and sentence error rates of the pt-pt SR system, when working in dictation mode, are poor, where the same rates, for the dynamic context-free-grammar based command and control scenario, are more approximated to the en-us case.

As soon as the system starts running and all the information is loaded, the SR engine starts to analyse the audio from the microphone, trying to recognize any utterance. This analysis is done until the system is closed. Each time a recognition event is detected, a list of words is created and their audio stored in files, one word per file. If the pt-pt engine is used, it also provides a list of phones in the IPA format [10], corresponding to the phonetic transcription of the resulting recognized word or phrase. This is the information passed to the core, using a thread, so the engine keeps recognizing the words. It is not possible to
Fig. 1. Pipeline of the proposed system composed by the main modules: the speech processing tool and the plugin inserted in the visualization tool. Here, it is possible to see the different components, the main technology used in each, if any used, and the flow of information until the animation and sound are played. First the words are recognized in the speech processing module, then if in translation mode, the new words and audio are obtained, respectively by the translation services and the TTS, then the words are analysed to obtain their phones in the phrase processing module. After that the phones are aligned with the audio to get their timings and durations. Finally, the visemes, obtained from the phone-to-viseme map are passed to the visualization tool together with the timings, which forms the data required to play a speech animation.

obtain the phones duration and timing from the SAPI, therefore this information has to be obtained in the other modules.

Text Translation
To translate a sentence, the Microsoft Bing Translator [14] web service was chosen. This service was implemented with a SOAP interface, used by the module to make a translation request. The whole process works in an asynchronous way. When the message is translated, a method within the module is called, used to pass the information to the core.
Phrase processing
Splitting the words into their basic sound units, i.e. the phones, is the task of the phrase processing module. The splitting is done using a phonetic transcription dictionary (a phonetic lexicon) provided by Microsoft, both for English and Portuguese. The first is complete, therefore it is only necessary to split the sentence into its words and then get the phones from a dictionary structure that has the words as keys. The phones are represented in the Universal Phone Set (UPS) [12]. Regarding the Portuguese language, the provided lexicon is not complete. When a word or sentence is not available, the phones, obtained from the SR engine, are analysed and converted from the IPA format to UPS. The output is the list of phones in the UPS format.

Phone alignment
The phone alignment module is based on the statistical duration of each phone. The average duration is normalized so the sum of all phones’ duration, in a word, has the same length as its corresponding audio file. The pt-pt audio durations were obtained from the Microsoft, collected from the analysis of a database with 100 hours of speech. The en-us durations were obtained using Microsoft TTS, where a set of 500 words were spoken and the phones’ duration obtained. The time value is the average per phone. This module receives the audio and the phones and returns the durations per phone.

Speech Synthesis
Like the input module, the speech synthesis is also based on the SAPI [13], in this case the TTS engine is used, instead of the SR engine. It takes a text as an argument and generates the utterance from it. As it is only required to generate the audio, instead of playing the utterance, the TTS is used to directly create a file per words’ utterance. The output is the list of audio files.

Viseme Selection
The visual selection module contains the map between the phones and visemes, thus responsible for the viseme selection. The first step is to load a file that contains the mapping between the phones and the visemes. Currently this is a XML file that maps each phone to a viseme, one mapping per language. It is important to notice the phones are in the UPS format and the visemes’ names are the same as the ones used by the visualization tool. The selection of the visemes currently results of directly using the phones, obtained from the previous modules, and just get their corresponding viseme name. Their animation weights are always 1. The output is a list of the visemes and their weights grouped by word.

Communication
The Microsoft Message Queue (MSMQ) [15] was used to allow the communication between different programs. A system queue is created that all applications can use to send and read messages, thus being an easy way exchange information. A message containing the audio files, duration and weights is passed to this module that will place it in the MSMQ. One message per word is created.
3.2 Visualization Tool

The full automatic pipeline was implemented in the Serious Game Engine (SGE), which is the current engine used in Porto Interactive Center. Its first version was created under the LIFEisGAME project [11], whose prototype is completely implemented using the engine. It is undergoing active development and currently is in a beta phase. The SGE is a highly modular engine, consisting of a small core and a set of plugins/modules that can be added to perform specific tasks. The modules’ behaviour is defined by interface classes that establish the methods each module has to implement. A developer only has to define these methods and they will seamlessly integrate in the engine. To create the speech animation system, it was only necessary to add the plugin to the SGE and make it query the plugin interface for new speech animation data. Unfortunately after implementing the whole system, some problems arose from this choice, mainly related with the frame-rate. Using the SGE with allowed only visualization of the speech animation at 25 fps, which is not enough for speech, where 30/35 fps are the absolute minimum. Therefore, the plugin was also implemented within Maya, which even though it does not allow a full automatic pipeline, permits the creation of the animation with the desired frame-rate. Both cases, plugin for SGE and for Maya, are now briefly described.

**Plugin**

The plugin only has the function of retrieving the data from the message, i.e. it parses the data string and obtains the audio file, visemes, their durations and weights. The message itself is obtained from the communication module, which accesses the MSMQ and retrieves the data. It is the engine that controls when the plugin verifies the queue.

**Maya Extension**

A script for Maya was created that read the animation data from a csv, and from this created the time-line. To generate the file, the communication module, within the speech processing tool, was changed to store the data in a file, instead of the Microsoft Message Queue.

4 Results & Discussion

This system consists of automatically generating the speech animation from an utterance, therefore it is extremely important to validate the visual results. The animation was created using the Nene model, created by Xenxo Alvarez (technical director), that has 99 joints. A total of 80 visemes were created, 11 plus neutral for the minimum English viseme set and two sets for the Portuguese, 15 plus neutral for the minimum set and 51 plus neutral for the extended version, all done by Pedro Bastos (digital artist). The data necessary to create both minimum and complete pt-pt sets were obtained from Microsoft, while the en-us was based on the Preston Blair extended phoneme series [16], which defines the mouth, teeth and tongue shapes for most phones. The Preston Blair visemes are used in manual mouth animation, which in fact does not require all the visemes...
to generate the animation, however this is a problem when using a phone-to-viseme approach, therefore the missing visemes were mapped to the Portuguese minimum set. It took about 8 hours of work to complete the creation of all visemes.

The whole system was implemented in a laptop with an Intel Core 2 Duo T9600, 4Gb of memory and an ATI 4650 1Gb. The system was integrated in the SGE, thus obtaining fully automated speech animation system that works near real-time. If the recognition delay is ignored, it takes about one second from the recognition event until the animation data is on the SGE. All the audio was recorded using the Logitech ClearChat Pro USB Headset at 8KHz and mono stereo. The results can be seen in a video uploaded to youtube in http://www.youtube.com/watch?v=82jdi8HPZFo. The validation was done for two case studies, which are now described.

The system was tested only for two utterances, thus serving only as a proof of concept. The two utterances were "olá todos" and "animar a fala" (Figure 2). Both cases were animated using the Portuguese minimum phone set. The figures show the visemes for the same utterance's word. The audio for both utterances is also shown.

4.1 Qualitative Validation

The qualitative validation was done by showing the video of the character speaking to members of the research group, and, as expected, animating a speech by directly concatenating the visemes does not result in realistic speech animation. The animation is clearly over-articulated. It appears however, that the vowels visemes’s weight are somewhat near the correct value, at least for the ones that are more pronounced. This results from the fact that the louder the phones are spoken, the wider the visemes are. The speech animation was also integrated with basic head movements such as nod. And in this case the resulting animation looked as it was expected, where there was no interference between the speech animation and the head movements.

4.2 Quantitative Validation

The quantitative evaluation was done in terms of time required to animate a sentence. The system generated all the audio synchronization information, used to place the audio and the visemes in the correct position of Maya’s timeline, thus reducing the time to manually synchronize all the visemes. The weights had to be fine-tuned in both utterances by changing their weights and animation curves, used to control the variation of the weights. About 5 minutes per sentence were required to fine-tune the animation. Based on Xenxo’s previous experiences, he estimated it would take about 25/30 minutes per utterance. Thus, a reduction of 500/600 % of the time required to animate each utterance was achieved.
4.3 External Dependencies

The quality of the output animation is greatly influenced by each module and method used in it. Even though most modules contribute decisively to the animation generation, there are some factors that are not controlled by the implementation. It is also important to note that the animation itself also contributes to the quality of the results. These are referred as system’s external dependencies and are:

- **Speech recognition.** If the words are not recognized correctly, the resulting animation will be completely wrong, as the audio and animation will not be same. Furthermore, any audio analysis will produce non usable results.
- **Phone alignment.** The phone alignment module is based on the statistical duration of phones, which should be obtained from the analysis of considerable size databases. If not done correctly it may influence the audio-visual synchronization. Another possibility is the use of a force alignment technique, which requires the availability of a speech recognition engine for the language.
– **Translation.** The translation does not directly affect the animation quality. However, it can, and will influence the users’ expectations as non-coherent utterances may be animated.

– **Model and Animation.** The quality of the rig influences how the in-betweens from two poses are calculated. Furthermore, the blending of different poses contributes to the smoothness of the resulting animation. If not done properly, the viewer will find the animation awkward.

– **Visemes.** The system is model independent, which means the visemes can be created in any character, but for the animation to be correct, these visemes should resemble to the poses of a real person. When creating these, it is also necessary to be careful, as a viseme may be mapped to more than one phone, therefore they have to be created accordingly the mapping that was chosen. If the poses are too exaggerated the animation will look over-articulated even if the correct poses and weights are provided.

The main limitation of the system is, without a question, the absence of co-articulation simulation, which prevents the resulting animation from being fully realistic. It is important to note that co-articulation was never meant to be addressed, at least until now. There is, however an intrinsic limitation with the way phones are aligned, especially with English that is related to the statistical duration of each phone. The Portuguese set is capable of fully represent the statistical duration of the phones, however in English this is far from true, as the values were only obtained from a 500 words corpus of synthesized speech. Therefore, before trying to simulate the co-articulation it would be necessary to solve this issue.

It is also important note the choice of the model. As the model is not realistic, a person has lower sensitivity to errors that may occur, therefore it would be important, in future experiments, to use a human-like model to better understand the synchronization.

5 Conclusion

The purpose of this work was to create a fully automatic system capable of automatically generating the synchronization of the audio and the lips from a speech signal and at the same time support speech animation generation in different languages. That was successfully accomplished, however this is not enough to generate realistic speech animation, being necessary to simulate the co-articulation effect. This is the natural evolution of the system, where the goal is to change the visemes according to the audio.

Some work has already been made in this direction, where from the analysis of the audio it is already possible to change the weight of the viseme, e.g. when a person speaks louder the viseme is wider, i.e. its weight is higher. Such already reduces the problem of over-articulation, even though it is not enough. As a result, the next steps are to completely implement the weight variation system and to add a co-articulation model such as Cohen-Massaro’s with the purpose of
creating a fully automatic visual speech animation system capable of generating life-like speech animation with little to no fine tuning.

References
