From speech to 3D face animation

Peter Vanroose[1], Gregor A. Kalberer[2], Patrick Wambacq[1], and Luc Van Gool[1,2]

[1] K.U.Leuven, ESAT/PSI
[2] ETH Zürich, BIWI *
Peter.Vanroose@esat.kuleuven.ac.be

Abstract. In this paper we present a new method to animate the face of a speaking avatar —i.e., a synthetic 3D human face— such that it realistically pronounces any given text, based on the audio only. Especially the lip movements must be rendered carefully, and perfectly synchronised with the audio, in order to have a realistic looking result, from which it should in principle be possible to understand the pronounced sentence by lip reading.
Since such a system requires minimal bandwidth and relatively low computational effort, it could e.g. be used to transmit video conferencing data over a very low bandwidth channel, where the lip motion rendering is done at the receiving end, by only transmitting the audio channel, or in extremis even only an orthographic or phonetic transcription of the text together with precise phoneme timing information.

1 Motivation

Producing realistic synthetic face animation is a challenging task, which has been tackled since the early 70s. Obtaining sufficient realism is very difficult: to date, no system was able to pass a "facial animation Turing test". A variety of approaches can be found in literature [1, 3]. Most of these use the concept of 'visemes', which are the visual counterparts of phonemes: visemes are 'key mouth shapes', each one corresponding to one or more particular sounds (phonemes).

Our system first 'learns' the face expressions of all possible visemes and all viseme pair transitions from real 3D face dynamics, observed at frame rate for a triangulated mesh of about 10000 vertices on the faces of speaking actors. This learning process is time consuming, but is done offline and only once.

To obtain the necessary realism at the receiving end, it is not sufficient to just concatenate the sequence of visemes or viseme pairs corresponding to the spoken sequence of phonemes since that would give a 'jerky' effect, which is really unacceptable. Correct inter-viseme face expressions together with perfect timing are crucial to obtain the necessary coarticulation effects, which suddenly give the face dynamics a much more human appearance.

Correct timing alignment can be obtained by stretching the mouth shape sequence such that it exactly matches the audio. Finally, the obtained trajectory

* This work was partially supported the K.U.Leuven Research Fund
and by the E.U. 5th Framework project "MYTHE".

is post-processed by mapping it into 6D 'viseme space' (representing the 6 first components of an independent component analysis) and smoothing it by spline fitting.

2 Phoneme set and viseme selection

Phonemes are the basic acoustic units of a language. For every language, a particular set of phonemes is selected, which best suit that language. E.g., for English, we use 45 phonemes, plus one (/#/) to represent silence.

In contrast to phonemes, there is no general agreement on the choice of a viseme set. A single viseme may correspond to several phonemes, but on the other hand, due to coarticulation effects, a single phoneme (like /m/ in the word "mean") could be mapped to more than one viseme (as the /m/ in "moon" looks more 'rounded' than the /m/ in "mean").

We based our selection of visemes in English on the work of Owens [7] for consonants, with the difference that we only consider two variants ('rounded' and 'widened') of each consonant. For vowels, we use the viseme set of Montgomery and Jackson [6]. This gives a total of 20 visemes: 2×6 consonant groups (/p,b,m/, /f,v/, /t,d,s,z,th,dh/, /w,r/, /ch,j,sh,zh/ and /k,g,n,l,ng,h,y/), 7 for vowels (/i,ii/, /e,a/, /aa,o/, uh/, /@@/, /oo/ and /u,uu/) and one viseme for the 'neutral' position (silence).

It is important to note that our visual speech model combines the visemes with additional coarticulation effects, which highly adds to the realism. This is an important improvement over earlier methods [1–3].

3 Coarticulation

In order to obtain a naturally looking video sequence, not only the individual viseme key shapes but even more the transitions between these must be rendered as realistically as possible.

Just performing a (linear) interpolation between key frames gives an unnatural, jerky pronunciation. Therefore, inter-viseme mouth shapes must be 'learned' from real video data. Luckily, with the chosen set of visemes, it suffices to have a single, accurate model of every viseme pair transition, and then glue these together, thereby respecting the correct timing of each viseme.

The assumption which makes this a valid approach is the Markovian 'conditional independence' assumption between consecutive phonemes/visemes, which has proven successful in speech processing [4] and is still the underlying model of all current speech recognition systems, see below.

The extraction of inter-viseme mouth shapes is a time consuming process, but since it has to be done only once, on the training data, this can be seen as an offline processing which does not influence the real-time performance of our system.

In addition to the above, and if processing power in the rendering system is not a bottleneck, we post-process the obtained trajectory as follows. We first map it into 6D 'viseme space', which consists of the 6 first components of an Independent Component Analysis (ICA) of the deformations from the neutral position of the generic face model (see next section). Then this trajectory is made fluent by fitting splines to the viseme space coordinates of the viseme sequence.

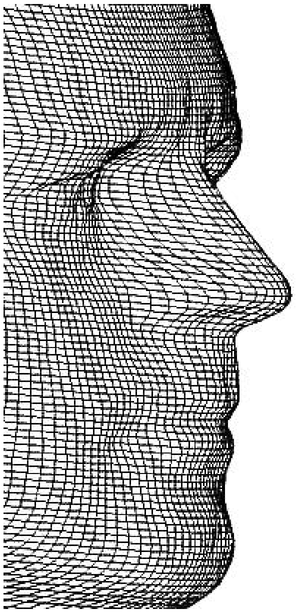
4 Precise 3D modelling of every viseme pair transition

To build a complete database of 3D face models for every viseme transition, we had to select a limited set of about 20 sentences in which every viseme pair is present in the form of two consecutive phonemes that map to these visemes. We did this for both English and Dutch.

Then several people pronounced these sentences, while at the same time the audio was recorded and their face was captured with a 3D system at 25 frames per second. To guarantee perfect synchronisation between audio and video, both streams were recorded on a single tape.

From these recordings, only the relevant video parts were cut out, viz. only the fragments corresponding to viseme pairs, with at most 5 fragments per viseme pair. These fragments were further processed in order to obtain a clean 3D model for each of the possible viseme transitions.

For the 3D recordings, the "ShapeSnatcher" system of Eyetronics was used [8]. It creates a 3D grid model as the one below (with additional texture) for in principle any surface, from a single picture. We mapped all obtained 3D face sequences onto a generic head (by using a common reference topology) in order to be able to train the characteristic mouth shapes for a specific viseme.



3D grid on avatar head

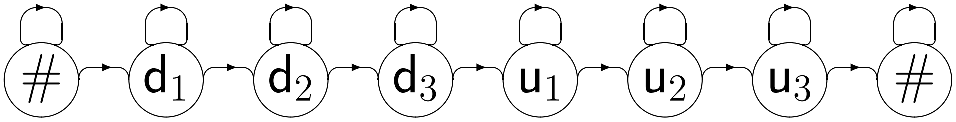
5 Modelling the acoustic signal

The viseme pair models can be learned from 3D video data, provided we have very accurate timing information from the audio input, and this both on the training data (audio and 3D video) and on the real-time audio input that is going to steer the virtual 3D face model.

The speech audio signal is first preprocessed to extract a compact 'feature vector' representation which still carries all essential information. The signal, sampled at 16 kHz, is segmented in chunks of 32 milliseconds with an overlap of 22 ms, i.e., the time resolution is 0.01 seconds. Each chunk thus contains 512 audio samples, from which the FFT transform is taken. From these 512 spectral values, 24 feature values are derived, the so-called Mel-scale cepstral coefficients, also including time derivatives [4].

The audio signal is now thus reduced to a trajectory through 24-dimensional feature space, sampled at 100 points per second. Most current state-of-the-art automatic speech recognition (ASR) systems use this representation to train an acoustic hidden Markov model (HMM) with 3 states per phoneme, where a transition is made once every 10 milliseconds.

For the English phoneme set with 45+1 phonemes, this implies training 138 × 138 transition probabilities from a sufficiently large database of 'annotated speech' (i.e., phoneme labelled and time-aligned), in conjunction with a probability model (usually a "union of Gaussians") for the conditional probabilities of a feature vector belonging to one of the HMM states. Since HMM states correspond to phonemes, not to observations (viz. the 24-dimensional feature vectors), this Markov model is called 'hidden'.



An acoustic model for e.g. a word with 2 phonemes like the word "do" /du/ is a Markov chain with 6 states, each phoneme having an 'initial' state, a 'middle' state and a 'final' state. The state transition probabilities for $\# \rightarrow d_1$, $d_1 \rightarrow d_1$, $d_1 \rightarrow d_2$, $d_2 \rightarrow d_2$, $d_2 \rightarrow d_3$, $d_3 \rightarrow d_3$, $d_3 \rightarrow u_1$, $u_1 \rightarrow u_1$, $u_1 \rightarrow u_2$, $u_2 \rightarrow u_2$, $u_2 \rightarrow u_3$, $u_3 \rightarrow u_3$ and $u_3 \rightarrow \#$ are trained by observing all those transitions in the training database, observing the corresponding feature vector sequences, and calculating the (empirical) conditional probabilities.

Note that acoustic models, just like the used phoneme set, are language specific.

6 Extraction of precise timing for viseme transitions

Given an audio signal, the acoustic model of an ASR system will thus be able to produce the sequence of phonemes that (according to the HMM) has the highest probability of matching the given audio. In order to obtain this optimal phoneme string the ASR must perform a precise segmentation of the audio, thereby assigning subsequent parts of the audio stream to each of the HMM states. Hence, as a side effect, the ASR system will produce precise timing information at a sub-phoneme level, with a resolution of 10 ms. It is this segmentation algorithm which will be used to obtain phoneme timing [5].

The alignment of the acoustic signal with the most likely state transition sequence, i.e., determining the path through the HMM with highest conditional probability given the feature vector sequence, is obtained by applying the *Viterbi algorithm*, originally designed for decoding convolution codes [9].

The performance of this segmentation is seriously improved if the phonetic transcription of the audio is known. This is especially true for the off-line training phase, since it ensures perfect viseme pair models. In this training phase we even used a manual phonetic transcription. In that case, the Viterbi algorithm reduces to a "Viterbi alignment" step, since only the duration of phonemes, i.e., the number of iterations where the HMM stays in the same state, is to be optimised.

It is less crucial (and also, in most applications, unrealistic) to have this information in the rendering system which is steered by the acoustic signal alone. But if we e.g. consider the most bandwidth-efficient mobile system in which only phonemes and their timing are transmitted, and the audio is generated by a text-to-speech system, then the phonetic transcription is indeed available. Note that the phonetic transcription can also be automatically derived from the orthographic transcription of the spoken sentences, given a phonetic dictionary.

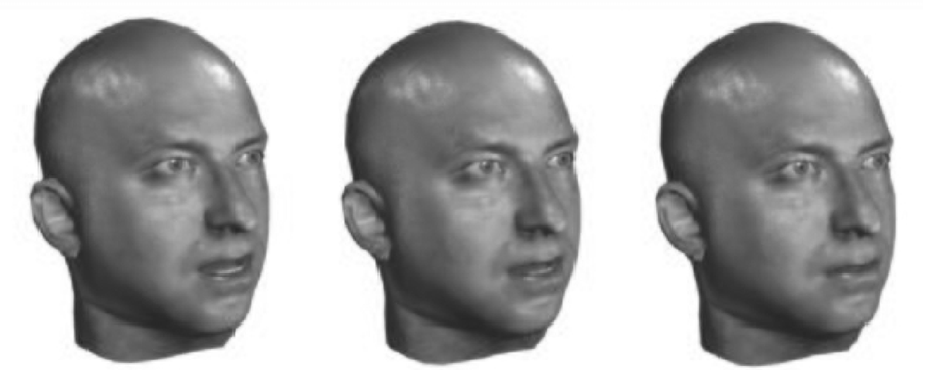
Applying the algorithm leads to the assignment of 3 timings (in multiples of 10 ms) to each phoneme in the transcription, one for each of the three states of the HMM model per phoneme. This sub-phoneme information turns out to be very useful in the viseme rendering, since it allows to have a more accurate timing of the mouth movements, which greatly adds to the realism.

7 Conclusions

Realistic face animation is still a challenge. We have tried to attack this problem via the acquisition and analysis of 3D face shapes for a selection of visemes, and by using very accurate viseme timing information, both in the offline training and in the client-side animation.

The animation is organised as a navigation through 'viseme space', as a concatenation of viseme pair transitions which could then be post-processed to an optimally smoothened trajectory. Given the input in the form of a spoken sentence, a face animation can be created fully automatically.

The proposed method yields realistic results, with more detail in the underlying, 3D models than earlier approaches.



Snapshots from rendered 3D sequence.

References

1. C. Bregler, M. Covell, M. Slaney, "Video rewrite: driving visual speech with audio", in *SIGGRAPH*, pp. 353–360, 1997.
2. M. Brand, "Voice Puppetry", in *Animation SIGGRAPH*, 1999.
3. T. Ezzat, T. Poggio, "Visual speech synthesis by morphing visemes", *International Journal of Computer Vision*, vol. 38, pp. 45–57, 2000.
4. X.D. Huang, Y. Ariki, M.A. Jack, *Hidden Markov Models for speech recognition*, Edinburgh University Press, 1990.
5. T.Laureys, K. Demuynck, J. Duchateau, P. Wambacq, "An Improved Algorithm for the Automatic Segmentation of Speech", in *Proc. 3rd Internat. Conf. on Language Resources & Evaluation (LREC02)*, Las Palmas, May 2002.
6. A. Montgomery, P. Jackson, "Physical characteristics of the lips underlying vowel lipreading performance", *Jour. Acoust. Soc. Am.*, vol. 73, pp. 2134–2144, 1983.
7. O. Owens, B. Blazek, "Visemes observed by hearing-impaired and normal-hearing adult viewers", in *Jour. Speech & Hearing Research*, vol. 28, pp. 381–393, 1985.
8. "ShapeSnatcher software", `http://www.eyetronics.com/`
9. A.J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm", *IEEE Trans. on Inform. Theory*, IT-13(2), pp. 260–269, 1967.