Speaker Independent Continuous Voice to Facial Animation on Mobile Platforms

Gergely Feldhoffer
Faculty of Information Technology, Pázmány Péter Catholic University, Budapest 1083, Práter u. 50/a, Hungary
E-mail: *flugi@itk.ppke.hu*

Abstract - *In this paper a speaker independent training method is presented for continuous voice to facial animation systems. An audiovisual database with multiple voices and only one speaker's video information was created using dynamic time warping. The video information is aligned to more speakers' voice. The fit is measured with subjective and objective tests. Suitability of implementations on mobile devices is discussed.*

Keywords - *facial animation, neural network, MPEG-4, DTW*

1. INTRODUCTION

Our group works on a voice to facial animation system for deaf people [1]. Our final target is a mobile device which supports this feature, allowing deaf customers to accept voice calls from any network, and understand it by lip-reading. Our previous system was a speaker dependent system, trained with only one person's voice. This paper will describe the development of a speaker independent but still continuous voice to facial animation system which can be implemented on mobile devices.

1.1. Voice to animation conversion

Voice to animation conversion systems (VACS) can be the area of descriptive research as speech inversion or applied research as applications for hearing impaired. Our project is on an application and has different purpose than speech inversion which tends to extract the exact state of speech organs from the voice signal. Our goal has two differences. Firstly, we try to get the state only the visible speech organs on the face, which is the source of lip-readable information.

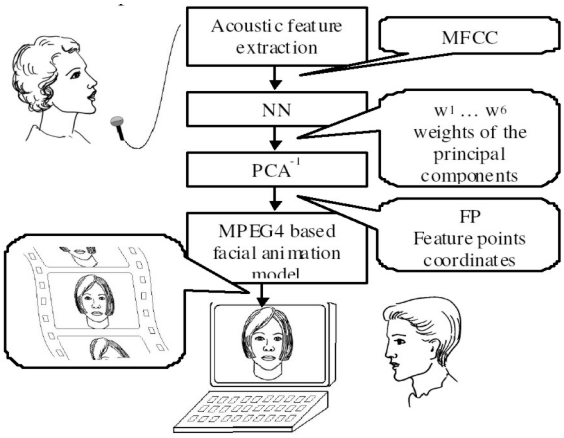

Fig. 1. Structure of a continuous voice to facial animation system

In the second place we do not want to restore the exact state of the speaker, rather we try to produce another face which is lip-readable even if the original speaker is not.

Continuous VACS means a system which uses only continuous methods on signals, and do not use any classifications, run-time database lookups, or any discrete values. For an example the SYNFACE project on KTH [3] is a discrete VACS using a module for speech recognition and another module for face synthesis. Both approaches have advantages. Discrete VACS can be trained on standard voice databases without video data, so there is no need to create new databases. Continuous VACS handles voice energy and rhythm naturally, it needs no a posteriori restoration of this information on the face model, and it needs no phoneme level labeling in the database.

1.2. Speaker dependency

The continuous VACS need an audiovisual database which contains audio and video data of speaking face.[2] The system will be trained on this data, so if there is only one person's voice and face in the database, the system will be speaker dependent. For speaker independency the database should contain more persons' voice, covering as many voice characteristics as possible. But our task is to calculate only one but lip-readable face. Training on multiple speaker's voices and faces results a changing face on different voices, and poor lip-readability because of the lack of the talent of many people.

We made a test with deaf persons, and the lip-readability of video clips is affected mostly by the person's talent, and any of the video quality measures as picture size, resolution or frame/sec frequency affected less. Therefore we asked professional lip-speakers to appear in our database.

For speaker independency the system needs more voice recording from different people. To synthesize

one lip-readable face needs only one person's video data. So to create a continuous VACS the main problem is to match the audio data of many persons with video data of one person.

2. SYSTEM DESCRIPTION

As it can be seen on Fig. 1. a running continuous VACS uses signal processing modules to extract audio feature vectors from audio data, applying the conversion with a machine learning method, decompress video data, and synthesize a face model. Training this system needs pairs of preprocessed audio and compressed video data. For this reason we are using the same window size in both modalities.

2.1. Audio preprocessing

The feature extraction starts with windowing. The length of a window depends on the frequency of the video camera which is 25 fps in our case; this means 40 ms long windows. Preemphasis is used, and FFT after Hamming window. We are using Radix-2 FFT for CPU efficiency, so the first 2^n element of the window is processed. The spectrum is mel-scaled to 16 bands and logarithm and DCT is applied. The result is the mel-scaled cepstrum, the MFCC.

2.2. Video processing

For the database the video recording is analyzed to extract the most expedient features of a face model from the point of view of lip-readability. We are using a subset of MPEG-4 standard feature points (FP) to describe the visible speech organs. The subset contains the inner and the outer contour of the mouth and reference points on the nose and on the chin. These features were extracted automatically with some level of manual tuning of parameters. The error ratio of the feature tracking is below 1% in the important phases of the recording. The tracking of the contour is based on color information, the errors are mostly on the inner contour, the outer contour is easier. [5]

The feature points are given in pixel coordinates on the video. This representation needs compression because of the great redundancy. We are using Principal Component Analysis (PCA) to compress this data. The 36 values of 18 FP-s are encoded with 6 coefficients on the PCA basis.

2.3. Neural network training

The synchrony of the audio and video data is checked by word " papapa" in the beginning and the end of the recording. The first opening of the mouth by this bilabial can be synchronized with the burst in the audio data. This synchronization guaranties that the pairs of audio and video windows were recorded in the same time, which gives an input and an output for a back-propagation neural network [4]. For the best result the neural network has to be trained on larger temporal scope of audio information. The mutual information between audio and video data was measured for different time shifts. 200 ms delay is advised. This is basically because of the speech process is a predicting mental process which moves the mouth according to not only the actual but the next approximately 200 ms of voice.[6]

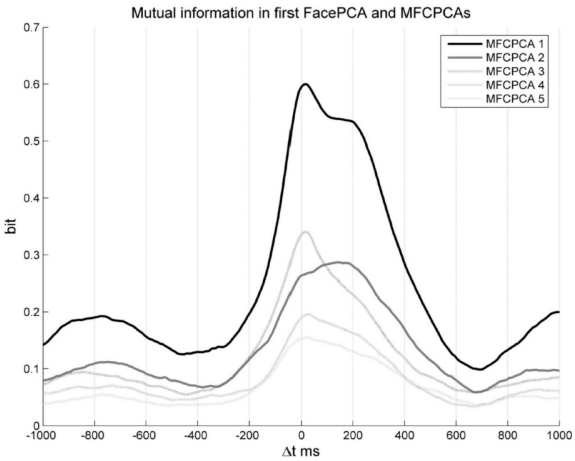

Fig. 2. Mutual information between the first PCA coefficients of the video and audio data

Therefore we are using a 5 element sized queue of audio feature vectors as the input of the neural network, and the corresponding PCA vector from the video data.

3. SPEAKER INDEPENDENCY

The described system works on well defined pairs of audio and video data. This is evident if the database is a single person database. If the video data belongs to a different person, the task is to fit the audio and the video data together.

The text of the database was the same for each person. This allows the aligning of audio data between speakers. We used the Dynamic Time Warping technique for this, which is a widely used method in speech recognition on small dictionaries. Usually for speech recognition purposes this is a distance estimation method using cumulative distance sums. We used it to extract the best match of the windows between the two audio data.

This matching is represented by index arrays which tell that speaker A in the *i* moment says the same as speaker B in the *j* moment. As long as the audio and video data of the speakers are synchronized, this gives the information of how speaker B holds his mouth when he says the same as speaker A speaks in the moment *i*.

With this training data we can have only one person's video information which is from a professional

lip-speaker and in the same time the voice characteristics can be covered with multiple speakers' voices.

3.1. Subjective validation

The DTW given indices were used to create test videos. For audio signals of speaker A, B and C we created video clips from the FP coordinates of speaker A. The videos of A-A cases were the original frames of the recording, and in the case of B and C the MPEG-4 FP coordinates of speaker A were mapped by DTW on the voice. Since the DTW mapped video clips contains frame doubling which feels erratic, all of the clips was smoothed with a window of the neighboring 1-1 frames. We asked 21 people to tell whether the clips are original recordings or dubbed. They had to give scores, 5 for the original, 1 for the dubbed, 3 in the case of uncertainty.

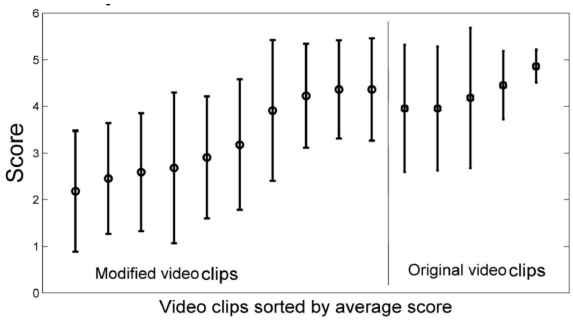

Fig. 3. Mean value and standard deviation of scores of test videos

As it can be seen on Fig. 3. the deviations are overlapping each other, there are better scored modified clips than original. The average score of original videos is 4.2, the modified is 3.2. We treat this as a good result since the average score of the modified videos are above the " uncertain" score.

3.2. Objective validation

A measurement of speaker independency is testing the system with data which is not in the training set of the neural network. The unit of the measurement error is in pixel. The reason of this is the video analysis, where the error of the contour detection is about 1 pixel. This is the upper limit of the practical precision.

40 sentences of 5 speakers were used for this experiment. We used the video information of speaker A as output for each speaker, so in the case of speaker B, C, D and E the video information is warped onto the voice. We used speaker E as test reference.

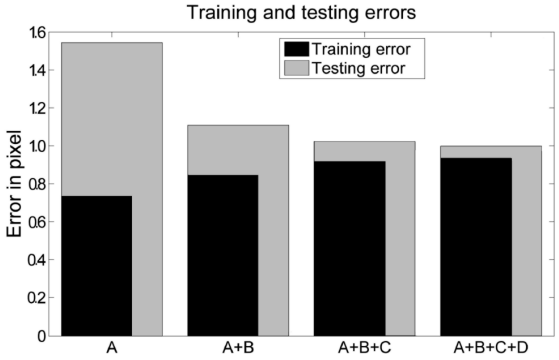

Fig. 4. Training with speaker A, A and B, and so on, and always test by speaker E which is not involved in the training set

First, we tested the original voice and video combination, where the difference of the training was moderate, the average error was 1.5 pixels. When we involved more speakers's data in the training set, the testing error decreased to about 1 pixel, which is our precision limit in the database. See Fig. 4.

4. DISCUSSION

The system should be implementable on mobile platforms. The audio preprocessing is O(n*log2(n)) where n is a function of the sampling rate. Computation time can be decreased by using a sampling rate which is enough for speech processing, as 16 kHz. This process needs little memory because of the possibility to operate in place with Radix-2 for both FFT and DCT.

To run the neural network, it can be solved with two matrix multiplications. The size of the matrices depends on the neuron numbers of the network. Using the network (see on Fig.4.) 80 nodes on the input layer, 40 nodes are in the hidden layer, and 6 nodes on the output layer gives only 3440 multiplications. The neural network needs also 3440 floating point data in the memory which is easily affordable.

Decompressing the video data from PCA values is one matrix multiplication with a 6x36 sized basis converting matrix, which occupies the same size in the memory.

Synthesis of the face model is the most time consuming part of the process. The speed of the application is limited by this part of the system. This module can be fastened by decreasing the number of vertices of the face model, but this can worsen lipreadability, and it influences the rendering speed just lightly. We are using a sparing model, see Fig 5.

The majority of the time complexity is the rendering. Some mobile devices support hardware implementation of OpenGL ES. Controlling the shape is faster but still slow. A face model in the memory uses 3 coordinates for each vertex and 3 integers on faces of the triangles. John has 346 vertices in 631

triangles. Time complexity of face motion is very high. Each vertex should be placed depending on the motion of feature points. This is 346*18 interpolation operation with smoothing. Smoothing is done by cosine window with different influence distances for each FP. For example the FP on the chin is moving the whole jaw almost uniformly, while a FP on the inner lip moves only 3-6 vertex with different weights depending on the distance.

All of the operations described above have to be done real-time for every voice signal window. As we measured, 12 frame/s is fast enough to achieve good lip-reading results on the screen of a mobile device.

The errors were given in pixel, and average errors were shown. In the practical use of VACS the maximum error is important also. This system has sparsely occurring relatively high errors due to misdetections during the video analysis, and rarely non-speech voice phenomena misleads the DTW. Also, there are problems with pure DTW because of the limit of the gradient. This allows only twice as fast or twice as slow warped signal on every location. There are situations where this limit is overrun, for example longer or shorter pause in between parts of sentences.

5. CONCLUSION

A speaker independent VACS is presented. Subjective and objective tests confirm the sufficient suitability of the DTW on training data preparing. It is possible to train the system with only voice to broaden the cover of voice characteristics. Most of the calculations are cheap enough to implement the system on mobile devices. The speaker independency induces no plus expense on the client side. The most problematic part is the implementation of the head model. A very basic head model as John still consumes more CPU than it would be affordable on an average open system 2nd generation mobile devices as Nokia Series 60 family for example. However on 3rd generation IP based video communication gives the possibility to implement this application in server-client architecture, which is already accomplished on PC. It is advised using a 2D head model instead, or implementing on 3rd generation client to accept any voice calls.

Speaker independency is an important feature of a VACS, and it is possible to change only the training phase.

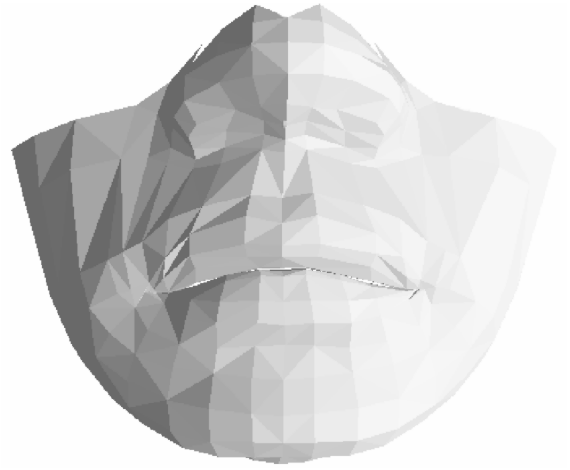

Fig. 5. Face model " John" with very low number of vertices

ACKNOWLEDGEMENT

We are thankful for the help of the members of SINOSZ, the Hungarian association of hearing impaired. I would like to thank Tamás Bárdi, Balázs Oroszi, György Takács and Attila Tihanyi for their help and comments.

REFERENCES

[1] G. Takács, A. Tihanyi, T. Bárdi, G. Feldhoffer, B. Srancsik, "Speech to Facial Animation Conversion for Deaf Customers", *EUSIPCO* Florence Italy, 2006.

[2] G. Takács, A. Tihanyi, T. Bárdi, G. Feldhoffer, B. Srancsik, "Database Construction for Speech to Lipreadable Animation Conversion", *ELMAR* Zadar, Croatia, 2006, pp. 151-154.

[3] B. Granström, I. Karlsson, K-E Spens: „SYNFACE – aproject present ation" *Proc of Fonetik 2002, TMH-QPSR*, 44: pp. 93-96.

[4] D. Anguita, G. Parodi, R. Zunino, "An efficient implementation of BP on RISC-based workstations", *Neurocomputing* Vol. 6 No. 1, 1994, pp. 57-65

[5] S. Gurbuz, "Real-time Outer Lip Contour Tracking for HCI Applications", *INTERSPEECH* Lisboa, Portugal, 2005 pp 1217-1220

[6] P. Scanlon, G. Potamianos, V. Libal, S. M. Chu "Mutual Information Based Visual Feature Selection for Lipreading", *Int. Conf. on Spoken Language Processing*, South Korea 2004