An Efficient Voice Driven Face Animation Method For Cyber Telepresence Applications

Gergely Feldhoffer and Balázs Oroszi
Pázmány Péter Catholic University
Budapest 1083 Práter u. 50/a, Hungary
flugi@itk.ppke.hu, oroba@digitus.itk.ppke.hu

***Abstract*—In on-line cyber spaces there are artificial bodies which imitate realistic behavior controlled by remote users. An important aspect is the realistic facial motion of human like characters according to the actual speech sounds. This paper describes a memory and CPU efficient method for visual speech synthesis for on-line applications using voice connection over network. The method is real-time, can be activated on the receiver client without server support. It is needed only to send coded speech signal and the visual speech synthesis is the task of the receiving client. This way deaf or hard of hearing people can lipread the transmitted audio speech. The animation rendering is supported by graphical accelerator devices, so CPU load of the conversion is insignificant.**

I. Introduction

Voice driven visual speech synthesis has a growing popularity in cyber telepresence applications. As of 2009 there are more video games on the market with the benefits of this technology.

The most popular use of visual speech synthesis is the real-time rendered pre-calculated facial animation. This meets all the requirements in an artificial world where the content of the voices is given by the designers, it is recorded with voice actors, and there is time to do all the calculations during production time. An example of this technology is in the title Oblivion or Fallout 3 from Bethesda Softworks[1] which uses the MPEG-4 based FaceGen[2]. However this approach is extendable to real-time applications as well by concatenative synthesis, we will see that it is not a really suitable solution.

In a real-time telepresence appliation the player activates the transmission, the client side records the voice in small chunks, and send it to the server which forwards it to the given subset of the players, teammates or any characters nearby. During active voice transmission the visual feedback on the receiver client side is some visual effect of the character, like an icon, a light effect, a basic or random facial motion. An example of basic facial motion is in the Counter-Strike Source, where the momentary voice energy is visualized by the movement of the jaw. Our solution is a replacement of this with improved quality, allowing even lipreading.

II. Overview

A. Real-time or pre-calculated motion control

In case of production time methods all of the audio content is available in advance. A typical example of this starts from screenplay, and the voice records are based on the given text. There are solutions to extract phoneme string from text, and to synchronize this phoneme string to the records like Magpie[3] for example. Voice synchronized phoneme strings can be used to create viseme string with visual co-articulation. The viseme is the basic unit of visual speech (Fig 2), practically the visual consequence of pronouncation of a phoneme. The viseme string with timing includes the visual information, and co-articulation methods has to form it into a natural visual flow. Viseme combinations were mapped for interactions as domination or modifying, and with this knowledge, viseme pairs or longer subsequences are used for the synthesis.

Also, during production time the speech signal is available as a whole sentence. This makes those methods usable which uses data for a given frame from the voice of next frames. This information definitely important for precise facial motion[4], [5]. Real-time methods are not allowed to use long buffers because of the disturbing delay.

One of the real-time approaches uses automatic speech recognition (ASR) system to extract phoneme string from the voice[5]. The benefit of this approach is the compatibility with viseme string concatenator methods by simply use ASR instead of manually extracted annotated phoneme string information. The ASR system can be trained on usual speech databases without visual data. The drawback is the time and space complexity of the recognition, and the propagation of the recognition errors, because of the falsely categorized phonemes or words.

Other way is the direct conversion which is simpler and faster but usually less accurate because of the lack of language dependent information. The proposed method is a direct conversion, which may be less precise but can be calculated real-time. A benefit of direct conversion is the natural handling of emotional speech. From a phoneme string it is hard to restore the emotional content. Directly from the sound the learning system can approximate these situations also.

B. Direct conversion

The basic idea of the direct conversion is that the connection between the shape of the mouth and the actual voice is basically physical. The relation is not bijective, one can make different voices with the same mouth state, but when an application targets voice to facial animation conversion, the task is not to restore the speakers mouth from the voice (which is the area of speech inversion) but to synthesize a facial state

978-1-4244-4641-4/09/$25.00 ©2009 IEEE

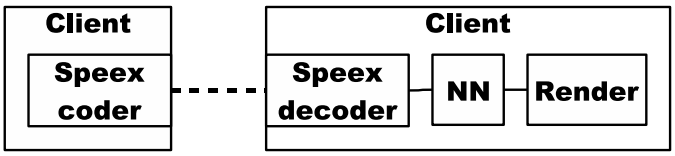

Fig. 1. Our real-time method: the visual speech animation parameters are calculated from the Speex coding parameters with neural network.

sequence with good articulation. Good articulation means high mutual information with the voice, in the best case lip-reading may be possible.

Direct conversion is based on a learning method. During production databases are created and trainings are done. This results an efficient runtime solution where the trained system just responses to the input (see Fig 1). The key issue of the direct conversion is to find an appropriate representation for the modalities. Both audio and video representation has to be compact and should contain all of the relevant information. In the next section the representations will be described.

III. Method

A. Face model

For a speaking head model there are two requirements: the artists should design the model easily, and it should have enough degree of freedom. For example in the game Counter-Strike Source the mouth motion has one degree of freedom, the position of the jaw, and it is directly linked to the energy of the signal. Although this behaves obviously artificial, this is numerically a fair approximation since visual speech PCA (Principal Component Analysis) factorization shows that the 90% of the deviation is in the first principal component which mainly shows the motion of the jaws[6]. In order to have a more sophisticated head model there should be more degree of freedom, which includes the horizontal motion of the mouth boundary or more.

In our works we used PCA based facial coordinates to represent a facial state. This representation have some nice properties as mathematically proven maximum compression rate along linear basises in dimension count. Each state is expressed in an optimal basis calculated from visual speech database. In this way a 30 dimensional facial data can be compressed into 6 dimension with only 2% error. As we visualize the calculated basis, the coordinates show motion components as jaw motion, liprounding, and so on. These are not visemes as visemes are not guaranteed to be orthogonals of each other. We used guided PCA in this case, including the most important visemes as long as it is possible.

For a designer artist it is easier to build multiple model shapes in different phases than building one model with the capability of parameter dependent motion by implementing rules in the 3D framework's script language. The multiple shapes should be the clear states of typical mouth phases, usually the visemes, since these phases are easy to capture by example. A designer would hardly create a model which is in a theoretical state given by factorization methods.

Therefore we need to give a facial animation control based on face states, and the designer can work with examples. The control of the facial animation can be the weights of the drawn shapes. Generally it is not true that every facial state can be expressed from any set of visemes, but there is an approximation of the states for a given viseme set, and depending on the size of this set, and so the degree of freedom, any level of accuracy can be reached (Table I). This approach may use more degrees of freedom than PCA based approach for the same quality, since the PCA is optimal in this point of view.

The rendering of the face is efficient. The graphical interfaces usually provide hardware accelerated vertex blending. There are more sophisticated approaches using volume conserver transformations [7] with slight increase of time complexity. These methods can be used to render our approach as well. Support for features like crinkling skin is out of our interest.

1) Viseme based decomposition: The video data is from a video recording of a talking person with fixed field of view. The head of the person was fixed to the chair to eliminate the motion of the whole head. The face of the person was prepared with marker points which were tracked automatically and corrected manually[6]. The position of the nose was used as origin, so every frame was translated to common frame. The automatic tracking was based on color sensitive highlight tracking with automatic quality feedback to help the manual corrections. There were 15 markers, placed on a subset of MPEG-4 feature points. The marker tracker results a vector stream in 2D pixel space. This representation is good for measurement, because no special equipment is needed for the recording, and also relatively good for estimation of quality since the generated animation will give the same data in best scenario. To achieve interchangeable metrics, pixel unit must be eliminated. This elimination is done by using the distribution of the given marker position as a reference to the position error. In this case the pixel units are eliminated from the result by transforming to a relative scale.

$$E(G) = \frac{\sum_{i=1}^{N} \sum_{j=1}^{f} \left| \vec{G}_j^i - \vec{S}_j^i \right|}{N f \sigma(\vec{S^i})} \qquad (1)$$

where N is the dimensionality of the visual representation, f is the total number of frames, G is the generated signal versus S signal and E is the estimated error value. S can be any linear representation of the facial state, given in pixels or vertices, or MPEG-4 FAP values.

Every facial state is expressed as a weighted sum of the selected viseme state sets. The decomposition algorithm is a simple optimization of the weight vectors of viseme elements resulting minimal errors. The visemes are given in pixel space. Every frame of the video is processed independently in the optimization. We used partial gradient method with a constraint of convexness to optimize the weights where the gradient was based on the distance of the original and the weighted viseme sum (Equation 1). The constraint is

Fig. 2. Visemes are the basic unit of visual speech. These are those visemes we used for subjective opinion tests in this (row-major) order.

a sufficient but not necessary condition to avoid unnatural results as too big mouth or head, therefore no negative weights allowed, and the sum of the weights is one. In this case a step in the partial gradient direction means a larger change in the direction and a small change in the remaining directions to balance the sum. The approximation is accelerated and smoothed by choosing the starting weight vector from the last result.

$$\vec{G} = \sum_{i=1}^{N} w_i \vec{V_i} \tag{2}$$

where

$$\sum_{i=1}^{N} w_i = 1 \tag{3}$$

The state G can be expressed as convex sum of viseme states V, which can be any linear representation, as pixel coordinates or 3D vertex coordinates.

The convexness guarantees that the blending is independent of the coordinate system. If the designer use unnormalized vertex coordinates, a weighted sum with more or less of weight sum of one can result translation and magnification of the head.

The results of this simple approximation are acceptable. The quality is estimated by pixel errors of the important facial feature points. The selection of important points is based on deviation, those feature points which are above the average deviation are chosen.

The head model used in subjective tests is three dimensional and this calculation is based on two dimensional similarities, so the phase decomposition is based on the assumption that two dimensional (frontal view) similarity induce three dimensional similarity. This assumption is numerically reasonable with projection.

TABLE I

ERRORS AS A FUNCTION OF VISEME NUMBER USED IN COMPOSITIONS ON IMPORTANT FEATURE POINTS. VISEMES ARE WRITTEN IN CORRESPONDING PHONEME CODES, EXCEPT "CLOSED" AND ʃ WHICH IS NOT A STANDALONE DOMINANT VISEME, IT IS USED AT THE BEGINNING OF THE WORD.

No.	Visemes	Error
2	*closed* ʌ	15.25%
3	*closed* ʌ ɪ	7.48%
4	*closed* ʌ ɪ ɔ	4.17%
5	*closed* ʌ ɪ ɔ ɛ	3.88%
6	*closed* ʌ ɪ ɔ ɛ ʃ*	3.48%

Note that the representation quality is scalable by setting the viseme count. This will make the resulting method scalable on client side.

B. Voice representation

Every voice processing method need to extract useful information from the signal. Those algorithms which use directly the sound pressure signal are called time domain, which uses Fourier transform or other frequency related filter banks are called frequency domain, and those which uses some (lossy) compressed input are called compressed domain methods.

Those applications, where voice driven facial animation can be a matter, use voice transmission. Voice transmission systems use lossy compression methods to minimize the network load. Therefore an efficient voice driven visual speech synthesizer should be a compressed domain method.

A speech coder attends to achieve best voice quality with reasonable sized data packets. This can be treated as a feature extracting method. The question is, what distance function can be used on the given representation? Is there an appropriate metrics what a learning system can approximate?

1) Speex coding: One of the most popular speech coder for this purpose is the Speex[8]. Speex uses LSP, a member of the linear prediction coding family. Linear prediction use a vector of scalars which can predict the next sample from the previous samples by linear combination.

$$x'_n = \sum_{i=1}^{N} a_i x_{n-i} \tag{4}$$

Where N is the size of the prediction vector. The optimal predictor coefficient vector for a given x can be calculated by Levinson-Durbin algorithm. This is short representation, but it is not suitable for quantization or linear operations as linear interpolation, consequently it is not directly used for voice transmission or facial animation conversion. Hence Speex uses LSP which is a special representation of the same information but capable to linear operations, for example the LSP values are linearly interpolated between the compressed frames of Speex.

For LSP coding, instead of storing the predictor vector a we treat it as a polynomial, and store the roots. The roots are

guaranteed to be inside the unit circle of the complex plane. To find roots, two dimensional search would be needed, so to avoid this we use a pair of polynomials which are guaranteed to have all the roots on the unit circle, and the mean of the pair is the original root, so one dimensional search is enough.

$$PQ_z = \left\{ a_z \pm z^{-(N+1)} a_{z^{-1}} \right\} \tag{5}$$

$$LSP = \bigcup_{z \in \mathbb{C}} \{PQ_z = 0\} \tag{6}$$

This makes LSP more robust to quantization and interpolation than the predictor vector. Interestingly, PQ values are called vocal tract, with glottis open and closed, which are connected with the topic of audiovisual speech synthesis.

Lossy compression methods use quantization of values of a carefully chosen representation. LSP is a compact and robust representation, and Speex use Vector Quantization to compress these values. We modified the Speex decoding process to export uncompressed LSP values and the energy. This makes only 11 assignments and multiplications for scaling as an extra computational cost.

C. Neural network training

The data is from an audiovisual recording of a professional lip-speaker. The recording contains 4250 frames. The content is intended for direct voice to visual speech conversion testing for deaf people, it contains numbers, months, etc. The language is Hungarian. The network is a simple straightforward error-backpropagation network with one hidden layer.

1) Audio: The audio recording is originally 48kHz, and it is downsampled to 8kHz for Speex. We used the modified Speex decoder to extract LSP and gain values to train neural networks as input. There are values for each 20 ms window. LSP has values in $[0, \pi]$, and the neural network use the $[-1, 1]$ interval, so scaling was applied.

2) Video: The target of the neural network is the viseme weight vector representing facial state. As the original recording is 25 frame per second, and the audio data from Speex uses 20 ms windows, the video data was interpolated from 40 ms to 20 ms frame interval. We used linear interpolation as it not violates convexness. The decomposition weight values are in the range of $[0, 1]$ which is in the neural networks $[-1, 1]$ interval, so no scaling was applied.

3) Neural network usage: The resulting network is intended to be used directly in the host application. The trained network weights can be exported as a static function of a programming language, for example C++. This source code can be compiled into the client. This function is called with the values exported from the modified Speex codec. The returning values is applied directly for the renderer. With this approach runtime overhead is minimal, no file readings or data structures are needed. The generated source code can be created at speech interested laboratories, the application developers just use the code.

D. Implementation issues

The method can be implemented as a feature on the client, on receiver side. The user may turn on and off the method since the calculation are performed on the receiver clients CPU. There is no extra payload on the network traffic.

The CPU cost of the calculation is 200-400 multiplications depending on the hidden layer size and the degrees of freedom. The space cost of the feature is the multiple shapes of the head models, which depends on the given application, how sophisticated head models are used in it. The space cost is scalable by setting the viseme set, the more head models the better approximation of the real mouth motion.

The head models can be stored on video accelerator device memory and can be manipulated through graphical interfaces as OpenGL or Direct3D. The vertex blending (weighted vector sum) can be calculated on the accelerator device, it is highly parallel since the vertices are independent.

E. Speaker dependency

The method can be extended towards speaker independence. The raw method use a neural network trained for coherent audio and video data. For practical reasons the speaker must be a good articulator (professional lip-speaker for the hearing impaired), which narrows the possibilities, and on the other hand the voice database should contain wide selection of speakers. This problem can be solved with mixed data pairs: video data from a good articulator, and audio data from anybody else. A method is published to arrange the data series to each other correspondingly [9]. This is to be done on production time. The working audio to visual speech conversion suffer no extra cost at runtime with this feature.

IV. Results

Training and testing set was separated, and during the first 1‘000‘000 epochs (training cycles) of training the error of the testing set still decreased (Fig 3). Depending on the degrees of freedom the results are 1-1.5% of average error. Our former measurements gave sufficient intelligibility results at this level of numeric error. This shows that usable training error level can be reached before overtraining even with relatively small databases.

The details of the trained system response can be seen on Fig 4. The main motion flow is reproduced, and there are small glitches bilabial nasals (lips not close fully) and plosives (visible burst frame). Most of these glitches could be avoided using longer buffer, but it cause delay in the response.

Subjective opinion test was done to evaluate the voice based facial animation with short videos. Half of the test material was face picture controlled by decomposed data and the other half by facial animation control parameters given by the neural network based control data from original speech sounds. The opinion test included from 1 to 5 degrees of freedom of control parameters. Each control source and degrees of freedom combination was represented in 8 short video, 2 of them pronounced numbers 0-9, 2 of them numbers 10-99, 2

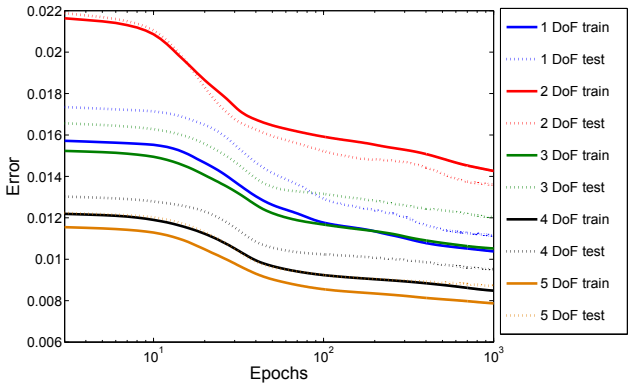

Fig. 3. Training and testing error of the network during training. The error is the average distance between the weight from the video data and the calculated from Speex input.

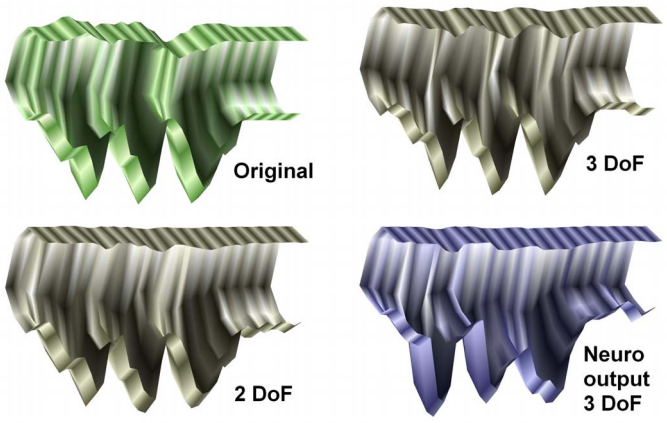

Fig. 4. Examples with the hungarian word ”Szeptember”, it’s very close to english ”September” except the last e is also open. Each figure is the mouth contour in the time. The original data is from a video frame sequence. The 2 and 3 DoF are the result of decomposition. The last pixture is the voice driven synthesis.

with names of the months and 2 with the days of the week. This makes 80 videos.

Test subjects were instructed to evaluate harmony and naturalness of the connection of visual and audio channels. Score 5 for perfect articulation, score 3 for mistakes and score 1 for hardly recognizable connection. The results are interesting since after the second degree of freedom the evaluation is near to constant while numerical error halves between 2. and 3. degree. The possible explanation of this phenomena can be the simpleness of our head model used for scoring. The tongue and the teeth was not independently moved in the videos, the more degree of freedom was used only to approximate the mouth contour more precisely which may was precise enough already at lower degrees of freedom.

Higher target complexity induce the neural network converge slower, or not at all. But as we increase the degrees of freedom, the neural network’s error decreases from the 2. degree.

The results of the opinion test show that the best score/DoF

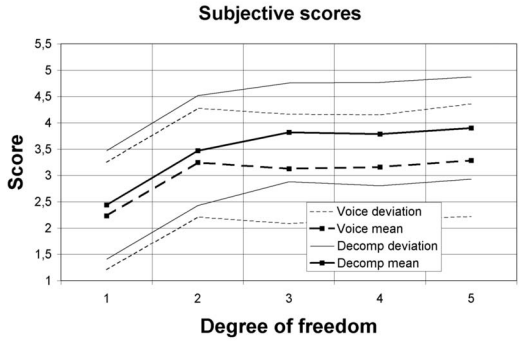

Fig. 5. Subjective scores of the decomposition motion control and the output of the neural network. There is a significant improvement by introducing a second degree of freedom. The method’s judgment follows the database’s according to the complexity of the given degree of freedom.

rate is at the 2 DoF (Fig 5), in fact the highest numerical error. These results show that the neural network may train to details which are not very important to the test subjects. As the decomposition is based entirely and only on mouth contour, it may be not that important. Using correct teeth visibility or tongue movement may improve the results, but in this test we were unable to try this because of the lack of markers on these facial organs. This problem is in the decomposition phase since in the synthesized face we have these facial organs and control them actively, but the control is inaccurate. If the decomposition would be affected by more information, this could be corrected. Active shape modeling or other advanced techniques may improve the decomposition material.

The main consequence of the subjective test that two degrees of freedom can give sufficient quality for audiovisual speech, and the proposed method can give the control parameters in this quality from the voice signal.

V. Conclusion

The presented method is efficient as the CPU cost is low, there is no network traffic overhead, the feature extraction of the voice is already performed by voice compression, and the space complexity is scalable for the application. The feature is independent from the other clients, can be turned on without explicit support from the server or other clients.

The quality of the mouth motion was measured by subjective evaluation, the proposed voice driven facial motion shows sufficient quality for on-line games, significantly better than the one dimensional jaw motion.

Let us note that the system does not contain any language dependent component, the only step in the workflow which is connected to the language is the content of the database.

VI. Acknowledgments

The authors would like to thank the help of György Takács, Attila Tihanyi, and Gergely Soós for their valuable questions

and comments and the Pázmány Péter Catholic University for the studio.

REFERENCES

[1] http://www.bethsoft.com.
[2] http://www.facegen.com.
[3] http://www.thirdwishsoftware.com/magpiepro.html.
[4] Gergely Feldhoffer, Tamás Bárdi, György Takács, and Attila Tihanyi. Temporal asymmetry in relations of acoustic and visual features of speech. In *15th European Signal Processing Conf.*, Poznan, Poland, 2007.
[5] J. Beskow, I. Karlsson, J. Kewley, and G. Salvi. Synface - a talking head telephone for the hearing-impaired. *Computers Helping People with Special Needs*, pages 1178–1186, 2004.
[6] György Takács, Attila Tihanyi, Tamás Bárdi, Gergely Feldhoffer, and Bálint Srancsik. Speech to facial animation conversion for deaf customers. In *4th European Signal Processing Conf.*, Florence, Italy, 2006.
[7] J. P. Lewis, Matt Cordner, and Nickson Fong. Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 165–172, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.
[8] Jean-Marc Valin and Christopher Montgomery. Improved noise weighting in celp coding of speech - applying the vorbis psychoacoustic model to speex. In *120th Convention AES*, Paris, France, 2006.
[9] Gergely Feldhoffer. Speaker independent continuous voice to facial animation on mobile platforms. In *49th International Symposium ELMAR*, Zadar, Croatia, 2007.