3D Image Analysis and Synthesis, pp. 51-56, Erlangen, November 1997.

# Speech Driven Synthesis of Talking Head Sequences

**Peter Eisert, Subhasis Chaudhuri [*], and Bernd Girod**

Telecommunications Laboratory, University of Erlangen,
Cauerstrasse 7, D-91058 Erlangen, Germany
Email: {eisert, sc, girod}@nt.e-technik.uni-erlangen.de

## Abstract

In this paper we present a method for generating talking head sequences directly from speech signals. A neural network is used to derive MPEG-4 facial animation parameters related to a person's mouth shape with low computational complexity. For the training of the network we use estimated parameters from an iterative and linear algorithm that uses 2D point correspondences of marker positions in video sequences. Having estimated the facial parameters we can render an animation of a speaking arbitrary person. Experimental results show that the appearance of the animated talking person looks natural.

## 1 Introduction

Research on video and speech processing is usually done independently. However, there is a high correlation between both modalities that is exploited in applications like visual speech recognition and speech driven lip motion synthesis. In the latter the animation of the lips can typically be derived either from the sampled speech signals [1, 2, 3] or from text [1, 4, 5]. This kind of system finds its use in multimedia applications, as automatic story teller [4], user friendly computer interface or video coding of head and shoulder scenes [1].

In this paper we propose a method for animating a 3D head model of a talking person directly from speech. We estimate the MPEG-4 SNHC facial animation parameters (FAPs) using a feed-forward neural network. It is not our intention to reconstruct the individual mouth motion characteristics of a specific person but to generate a natural looking animation synchronized with the given audio signal. We compute LPC coefficients and an energy measure which are both used as input for the neural network. This network returns directly the facial animation parameters allowing the use in real-time applications.

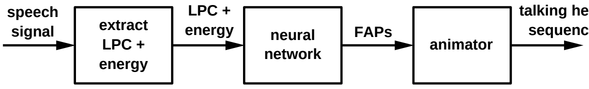

Figure 1: Schematic diagram.

The paper is organized as follows. First, the head model used for the rendering of the talking head is described. We then discuss a method for the estimation of the facial animation parameters from a video sequence. These estimates are needed for the training of the neural network that is described in section 4 together with the determination of the facial parameters from speech signals. Finally, experimental results for this approach are shown in section 5.

## 2 Modeling of the Head

For the animation of the talking head we need a 3D model that defines the appearance of the speaking person. We use a generic head model [6] that can easily be adapted to an individual described by a 3D laser scan. The topology of the 3D mesh and the modeling of the facial expressions do not change, which gives us the possibility to exchange the modeled person by another one without retraining of the neural network or modifying the estimation of the facial animation parameters.

Like other well-known head models [7, 8] our model is a 3D wireframe model (Figure 2). Texture is mapped onto the mesh to obtain a natural appearance (Figure 3). To simplify the model-

[*] Visiting Humboldt Fellow from the Indian Institute of Technology, Bombay.

ing of surface deformation caused by facial expressions, the surface is constructed using second order triangular B-splines [9]. With this new spline scheme that allows local mesh refinements, the surface is defined by a small number of control points. Instead of moving all vertices of the mesh, only the positions of the control points are changed in order to achieve local deformations or motion of the shape. The positions of the vertices are defined by the basis functions of the splines which model the facial tissue realistically [10].

For the specification of the facial expressions we adopt the parameterization of the MPEG-4 SNHC that is currently being standardized [4]. This scheme specifies 68 different facial animation parameters like 'open jaw' or 'stretch left cornerlip' and the facial expressions of a person are generated by superposition of 68 action units. For the animation of the head model, changes of facial animation parameters are transformed into translational or rotational movements of control points that determine the deformation of the surface.

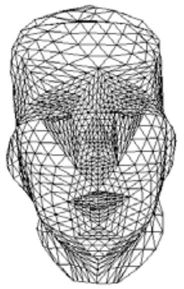

Figure 2: Wireframe model of the head.

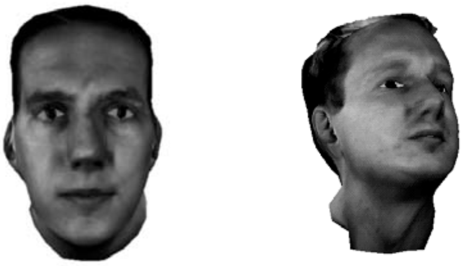

Figure 3: Head model for two different persons.

# 3 Estimation of Facial Animation Parameters from Video

The training of the neural network requires target vectors of the facial animation parameters that have to be learned. We must therefore estimate the facial animation parameters also from video sequences. However, this has to be done only once for the training of the network, which allows doing this under controlled conditions. Markers are used to determine 2D point correspondences in the video frames. From the point correspondences between two successive frames we estimate the changes in facial animation parameters using a linear iterative algorithm. Eleven parameters are currently estimated. For the compensation of global head motion three translational and three rotational parameters are used. The remaining five parameters (open jaw, raise left cornerlip, raise right cornerlip, stretch left cornerlip, stretch right cornerlip) specify the lip and mouth area and are used for training the network.

## 3.1 Determination of the Marker Positions

The marker positions are determined from the color images by a simple color conversion and thresholding operation. We attach 22 green markers in the face as shown in Figure 4 and record the video sequences. In the video frames a pixel is classified as a marker pixel, if

$$\frac{g}{r+g+b} > \Theta_1 \text{ and } r+g+b > \Theta_2 \qquad (1)$$

is fulfilled, where r, g and b are the red, green and blue components of the pixel color and $\Theta_1$ and $\Theta_2$ are two thresholds. After this operation, connected regions smaller than five pixels are removed leading to an image as shown on the right hand side of Figure 4. The exact marker position is then calculated from the center of gravity of the marker images. In subsequent frames the markers are searched in a small range around the previously determined positions.

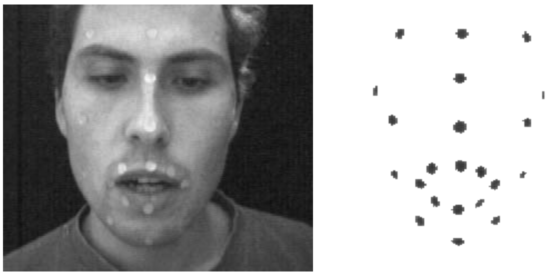

Figure 4: Camera frame of a training sequence (left) and extracted marker regions (right).

### 3.2 Animation Parameters from 2D Correspondences

For the estimation of facial animation parameter changes between two successive frames we use a modified version of the method proposed in [11]. Instead of using the optical flow constraint equation, the approach has been adapted for the use of 2D point correspondences. We can set up an explicit linear function for each marker that defines its displacement vector as a function of the unknown facial animation parameters (FAPs). This relationship differs for each marker due to local deformations caused by facial expressions and is generated combining several transformations as shown in Figure 5. Changes in the facial animation parameters $a_k$ are transformed by the head model into control point movements according to

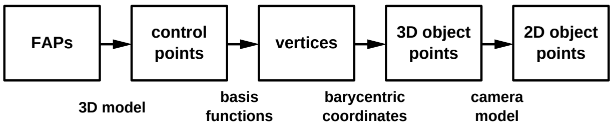

Figure 5: Transformation chain from animation parameters to 2D point correspondences.

$$\mathbf{c}' - \mathbf{c} = \sum_k a_k \mathbf{d}_k, \tag{2}$$

where $\mathbf{c}'$ and $\mathbf{c}$ are the 3D positions of the control point in the current and previous frame and $\mathbf{d}_k$ the 3D direction vector associated with parameter $a_k$. The position of the control points then defines all vertex locations $\mathbf{v}$ by

$$\mathbf{v} = \sum_{i \in I} N_i \mathbf{c}_i, \quad \text{such that} \quad \sum_{i \in I} N_i = 1, \tag{3}$$

with $N_i$ being the precalculated basis functions of the triangular B-splines. The 3D marker position $\mathbf{x} = [x\ y\ z]$ is finally determined by the barycentric coordinates $\lambda_m$ in the surrounding triangle

$$\mathbf{x} = \sum_{m=0}^{2} \lambda_m \mathbf{v}_m. \tag{4}$$

All these transformations are linear and can be combined into a single one with a new transformation matrix $\mathbf{T}$ and the vector $\mathbf{a}$ of facial animation parameters

$$\mathbf{x}' - \mathbf{x} = \mathbf{T} \cdot \mathbf{a}. \tag{5}$$

The relationship between 3D points and their corresponding 2D projections $[X\ Y]$ is given by the camera model. In our case we use a simple perspective projection according to

$$\begin{aligned} X &= X_0 - f_x \frac{x}{z} \\ Y &= Y_0 - f_y \frac{y}{z}. \end{aligned} \tag{6}$$

The internal camera parameters $X_0$, $Y_0$, $f_x$ and $f_y$ are obtained from a camera calibration and define the position of the optical axis on the CCD-sensor, the viewing angle and the aspect ratio of the pixels. The camera model and its associated coordinate systems are depicted in Figure 6. The camera model (6) is then used to project the 3D points of equation (5) into the 2D image plane. After first order approximation we obtain two linear equations for each marker

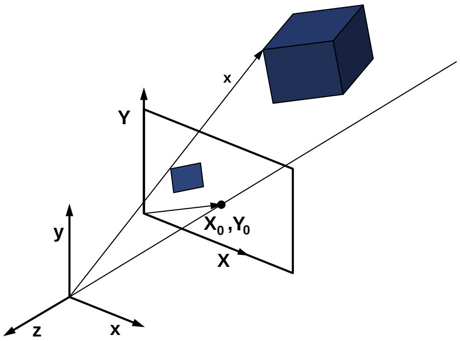

Figure 6: Camera model.

$$\begin{aligned} X' - X &\approx -\frac{1}{z}\left(f_x \mathbf{t}_x + (X - X_0)\mathbf{t}_z\right)\mathbf{a} \\ Y' - Y &\approx -\frac{1}{z}\left(f_y \mathbf{t}_y + (Y - Y_0)\mathbf{t}_z\right)\mathbf{a}, \end{aligned} \tag{7}$$

where $\mathbf{t}_x$, $\mathbf{t}_y$ and $\mathbf{t}_z$ are the three row vectors of matrix $\mathbf{T}$. $[X'\ Y']$ is the extracted 2D marker position in the actual frame and $[X\ Y]$ is calculated by projecting the 3D coordinate $\mathbf{x}$ of the previous frame into the 2D domain. The resulting overdetermined linear set of equations is then solved for the unknown animation parameters in a least-squares sense. The solution is, however, not exact due to the linearizations in equation (7). Therefore, the procedure is repeated iteratively with motion compensation after each step until the estimates for the facial animation parameters converge. On average, 2.1 steps are needed to get an accuracy of better than 0.1 % of the maximum value.

# 4 Estimation of Facial Animation Parameters from Speech

The head model in section 2 is animated by facial animation parameters according to the MPEG-4 SNHC syntax. These parameters are derived directly from an audio signal of a speaking person. Five such parameters (Table 1) that are related to mouth and lip movements are used. Other facial expressions like eye blinking or global head movements can be added to increase the realism of the animation. For the joint estimation of the five animation parameters we use a simple three layer feed-forward neural network trained by backpropagation. Because of the low computational complexity of such a network, we can determine the parameters from the speech signal in real-time. The audio signal is preprocessed leading to eleven input parameters for the network. First, the speech signal $s(t)$ is divided into non-overlapping blocks of length T=33.33 ms corresponding to a frame rate of 30 Hz. The energy of the signal in each block i of length M

$$e_i = \frac{1}{M} \sum_{j=Mi}^{(i+1)M-1} s^2(j) \tag{8}$$

is calculated together with the first 10 LPC coefficients of the signal in the block. These eleven input parameters are fed into the neural network that returns the estimates for the five animation parameters (Figure 7).

| FAP number | FAP name |
|---|---|
| 3 | open jaw |
| 6 | stretch left cornerlip |
| 7 | stretch right cornerlip |
| 12 | raise left cornerlip |
| 13 | raise right cornerlip |

Table 1: Facial animation parameters (FAPs) that are estimated from the speech signal.

Once the neural network is trained, only the speech signal is needed for the determination of the mouth shape. The training of the network, however, requires target vectors for the facial animation parameters that are derived from video sequences as shown in section 3. In MPEG-4 the animation parameters are measured as multiples of the facial animation parameter units (FAPU) that are derived from characteristical distances in the face. In our implementation the parameters are normalized to lie between -1 and 1. During the training which is performed with the backpropagation method the values are further divided by 2 to avoid a saturation of the output neurons.

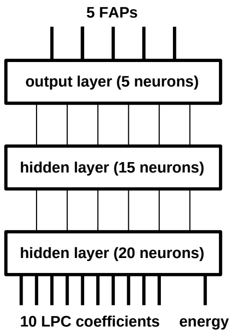

Figure 7: Structure of the neural network.

# 5 Experimental Results

The method proposed in the previous sections has been applied to several data sets. Audio signals of a talking person are sampled with 16 kHz. Simultaneously, video sequences of the person are recorded at 30 Hz in CIF resolution (352 x 288 pixels). The markers in the face of the person are tracked and the five facial animation parameters are estimated from the marker positions. These values are used to form target vectors for a neural network with two hidden and one output layer. Experiments showed that a size of 20, 15 and 5 neurons, respectively, for the three layers is a good choice. The network is then trained with a random set containing about 2000 blocks (frames) of data (block length T=33.33 ms). After the training, the network is driven only by the audio signal. The resulting animation parameters are postprocessed before using them for the animation of the head model. In speech pauses, which are detected from the calculated energy, the animation parameters are slowly decreased to zero corresponding to a neutral expression. Otherwise, the LPC parameters from the background noise lead to an unrealistic behavior. Additionally, the sequence of facial animation parameters is low-pass filtered with the filter

$$h = \begin{bmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{bmatrix} \tag{9}$$

to remove high frequency components. These components arise, because the parameters are estimated for each block independently without considering the state of previous blocks leading to a simple structure for the neural network.

Figures 8 and 9 show the estimated parameters 'open jaw' and 'stretch left cornerlip' in comparison to the values extracted from the video signal that are used as reference. This is depicted for 100 frames of a sequence that was also in the training set. The same is done for speech signals that are not used for the training. For comparison reference values are again estimated from marker positions in the corresponding video sequences. The results can be seen in Figures 10 and 11. As expected, the estimates are worse but they are sufficiently accurate to generate a realistic looking sequence.

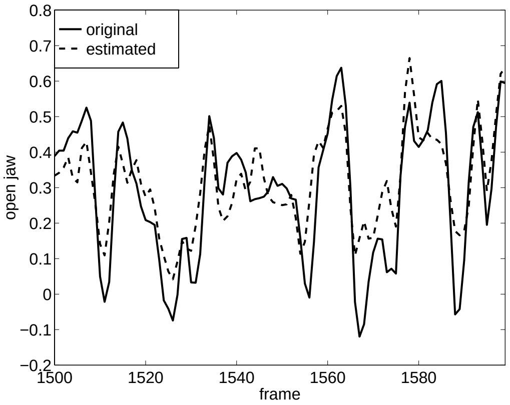

Figure 8: Original and estimated values of the unit 'open jaw.

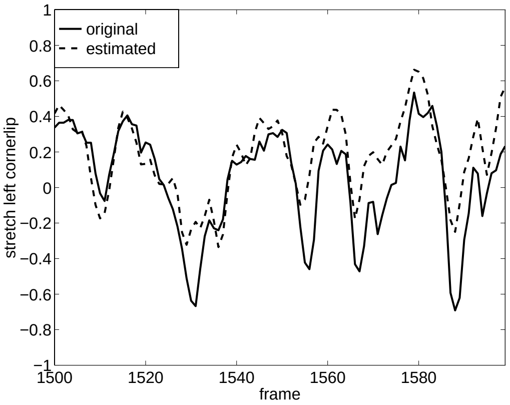

Figure 9: Original and estimated values of the unit 'stretch left cornerlip'.

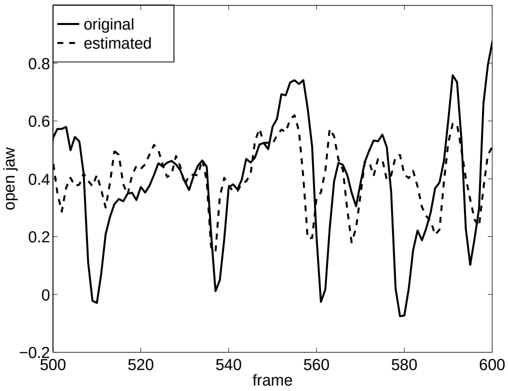

Figure 10: Original and estimated values of the unit 'open jaw' from a sequence not used for training.

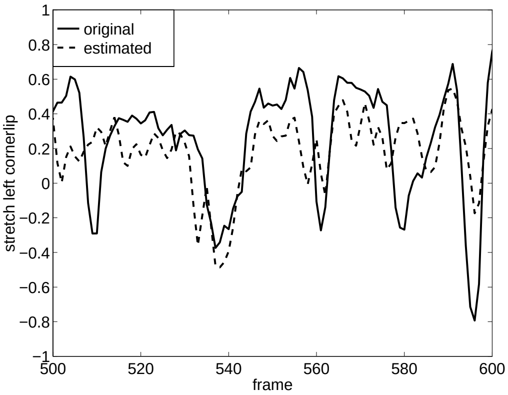

Figure 11: Original and estimated values of the unit 'stretch left cornerlip' from a sequence not used for training.

The head model is rendered with the estimated parameters leading to the images shown in Figure 12. On the left side, the synthetic images are generated from SNHC parameters determined from the markers in the video sequence. The global motion that is also extracted is compensated for a better comparison in the mouth area. On the right side, the corresponding frames, synthesized using facial animation parameters derived from the speech signal, are shown.

# 6 Conclusions

In this paper we have presented an algorithm that estimates SNHC facial animation parameters directly from speech signals using a feed-forward neural network. Additionally, the estimation of these values from marker positions in video sequences, which is necessary for the training of the network, is described. The experimental results show that natural looking video sequences can be synthesized with this low complexity algorithm suitable for real-time applications.

## References

[1] S. Morishima, K. Aizawa, and H. Harashima, “An intelligent facial image coding driven by speech and phoneme”, *ICASSP*, pp. 1795– 1798, 1989.

[2] R. R. Rao and T. Chen, “Exploiting audio-visual correlation in coding of talking head sequences”, *Picture Coding Symposium*, pp. 653–658, Mar. 1996.

[3] F. Lavagetto, S. Lepsøy, C. Braccini, and S. Curinga, “Lip motion modeling and speech driven estimation”, *ICASSP*, vol. 1, pp. 183–186, 1997.

[4] MPEG-4, *SNHC Verification Model 4.0, Document N1666*, Apr. 1997.

[5] C.-H. Cheung and L.-M. Po, “Text-driven automatic frame generation using MPEG-4 synthetic/natural hybrid coding for 2-D head-and-shoulder scene”, *International Conference on Image Processing*, Oct. 1997.

[6] P. Eisert and B. Girod, “Facial expression analysis for model-based coding of video sequences”, *Picture Coding Symposium*, pp. 33–38, Sep. 1997.

[7] F. I. Parke, “Parameterized models for facial animation”, *IEEE Computer Graphics and Applications*, pp. 61–68, Nov. 1982.

[8] M. Rydfalk, *CANDIDE: A Parameterized Face*, PhD thesis, Linköping University, 1978, LiTH-ISY-I-0866.

[9] G. Greiner and H. P. Seidel, “Modeling with triangular B-splines”, *ACM/IEEE Solid Modeling Symposium*, pp. 211–220, 1993.

[10] M. Hoch, G. Fleischmann, and B. Girod, “Modeling and animation of facial expressions based on B-splines”, *Visual Computer*, 1994.

[11] P. Eisert and B. Girod, “Model-based estimation of facial expression parameters from image sequences”, *International Conference on Image Processing*, Oct. 1997.

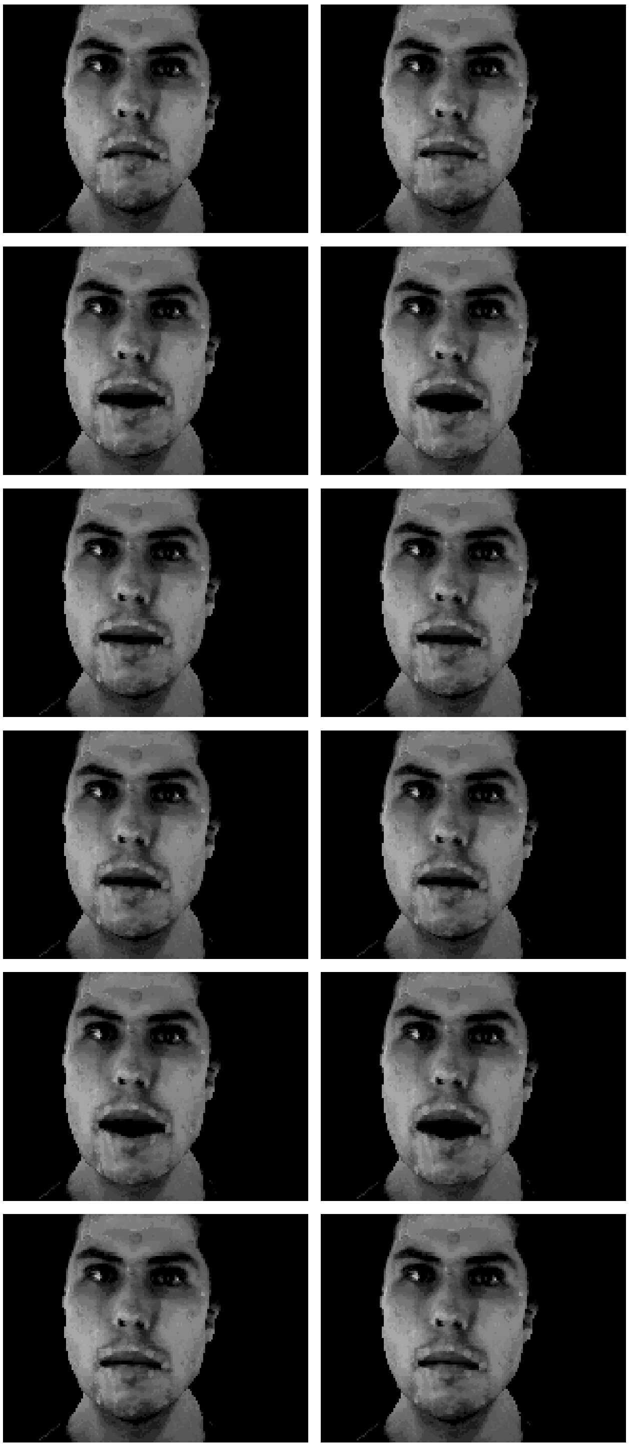

Figure 12: Left: synthesized facial expressions with parameters from the video (reference); right: speech-only expression synthesis. For an animated video sequence please point your WWW browser to *http://www.nt.e-technik.uni-erlangen.de/˜eisert*.