Multimodal Speaker Identity Conversion
- *Continued* -

Z. Inanoglu (#), M. Jottrand (+), M. Markaki (*), K. Stankovic (°), A. Zara (^), L. Arslan (~), T. Dutoit (+), I. Panzic (°), M. Saraçlar(~), Y. Stylianou (*)

(#) *University of Cambridge – UK,* (+) *Faculté Polytechnique de Mons – BELGIUM,*
(*) *University of Crete – GREECE,* (°) *University of Zagreb – CROATIA, ,*
(^) *France Télécom R&D – FRANCE,* (~) *Boğaziçi University – TURKEY*

Abstract— **Being able to convert a given the speech and facial movements of a given source speaker into those of another (identified) target speaker, is a challenging problem. In this paper we build on the experience gained in a previous eNTERFACE workshop to produce a working, although still very imperfect, identity conversion system. The conversion system we develop is based on the late fusion of two independently obtained conversion results: voice conversion and facial movement conversion.**

In an attempt to perform parallel conversion of the glottal source and excitation tract features of speech, we examine the usability of the ARX-LF source-filter model of speech. Given its high sensitivity to parameter modification, we then use the code-book based STASC model.

For face conversion, we first build 3D facial models of the source and target speakers, using the MPEG-4 standard. Facial movements are then tracked using the Active Appearance Model approach, and facial movement mapping is obtained by imposing source FAPs on the 3D model of the target, and using the target FAPUs to interpret the source FAPs.

Index Terms—**voice conversion, speech-to-speech conversion, speaker mapping, face tracking, avatar control.**

I. Introduction

THIS eNTERFACE'07 project is a continuation of a project started during eNTERAFACE'06 in Dubrovnik [1], in which we aimed at converting a given *source speaker* speech and facial movements into those of another (identified) *target speaker*. Such a conversion is typically based on some (separate) parametric models of the speech and facial movements for both speakers (Fig. 1). Two streams of (time-varying) parameters (one for the speech model, one for the face model) are first estimated from an audio-video file of the source speaker; some of these parameters are modified using *mapping functions;* the modified parameter streams are finally converted into an audio-video file which should hopefully be identified as originating from the target speaker.

The final quality of the conversion therefore depends on the quality of the rendering obtained by the parametric models and on the efficiency of the mapping functions, which both result from design choices.

Rendering quality can easily be estimated by copy-synthesis experiments: one takes an audio-video file as input, estimates parameters and performs rendering without modifying the parameters. Errors can be due to *modeling errors* (the model is not able of capturing all the details in the data) and/or to *estimation errors* (the model, when used for rendering, is capable of producing perfect copy-synthesis if it is fed with some optimal parameter stream, but the parameter estimation algorithm cannot find the best parameter values). This leads to a classical modeling trade-off: if too simple, a model lends itself to modeling errors; if too complex, it minimizes modeling errors but opens the doors to estimation errors.

Mapping should produce a sensation of identity conversion while not degrading too much the rendering quality obtained with copy synthesis. Here again, a trade-off usually has to be made: while applying smoothed mapping preserves copy-synthesis quality, it only partially produces identity conversion; conversely, applying hard mapping modifies the impression of identity but often significantly degrades quality [2].

In addition to begin dependent on the models and mapping methods it uses, speaker conversion systems are characterized by the type of data they are based on. Mapping functions are usually trained from *aligned* data between source and speaker, although a new trend is to design mapping functions from source and target speakers not uttering the same sentences.

Conversion approaches also differ by the assumptions they make on the size of the available data from the source and speaker, for training the mapping functions. Being able to train an efficient mapping function from limited data is more challenging (and often closer to real applications).

This report, as well as the source code for the software developed during the project, is available online from the eNTERFACE'07 web site: www.enterface.net/enterface07.

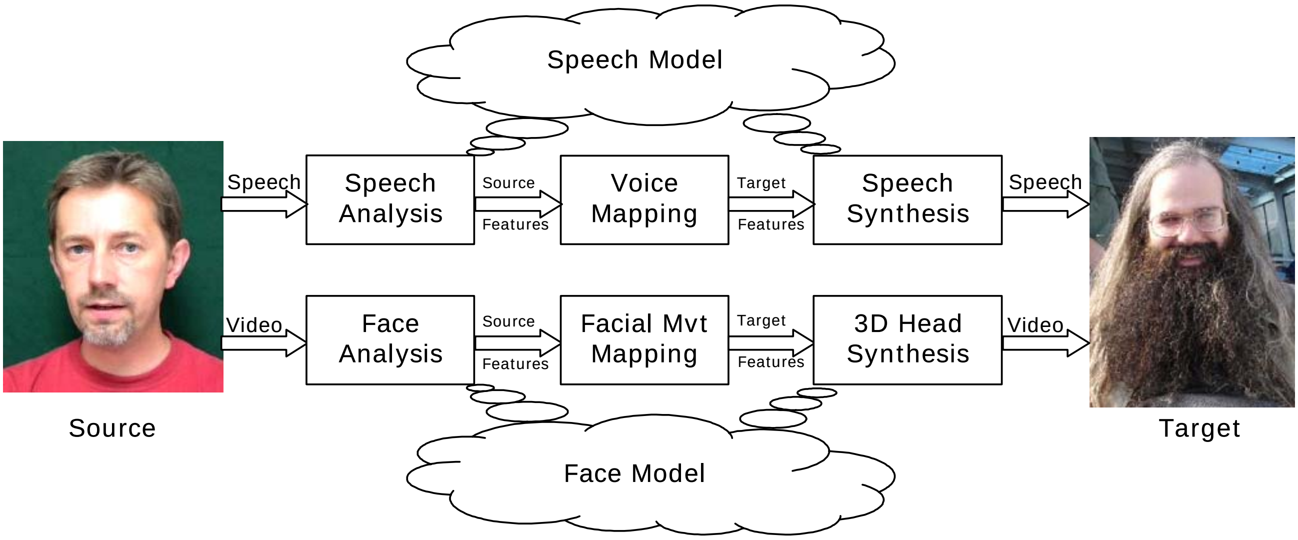


Fig. 1 Principles of speaker identity conversion, using source speech and facial movements, mapping them to target speech and face, and producing target-like speech and facial movement.

In this project, it is assumed that a large amount of aligned speech data can be recorded from both the source and target. As a matter of fact, even in such advantageous conditions, the state-of-the-art in voice conversion has not yet reached a level which would make it a widely usable tool for commercial applications. In contrast, we assume that only a photo of the target speaker is available. A typical application of this project is therefore that of a human actor controlling the speech and facial movements of a 3D character whose face and voice is well-known to the audience, and from whom a large amount of speech data is available.

The paper is organized as follows. Section II summarizes the speech model and mapping function we tested in [1] and examines the new choices made in this year's project. In Section III, we summarize the face model and mapping function (which has not changed from [1]). Experiments using these models and mappings (using the eNTERFACE06_ARCTIC[1] database created last year) are detailed in Section IV, followed by an analysis of our results, and perspectives, in Section V.

II. Speech Analysis, Mapping, and Synthesis for Voice Conversion

In [1], a number of choices had been made relatively to the modeling and mapping trade-offs mentioned in Section I.

Speech was modeled using Residue Excited Linear Prediction (RELP), which has the advantage of providing transparent copy synthesis, but consequently provided few means of mapping the glottal source signals from source to target speaker. A mapping function was then applied, frame-by-frame, on the vocal tract filter of the source, $1/A_s(z)$, based on Gaussian Mixture Models (GMM) of the source and target Mel-Frequency Cepstrum Coefficient (MFCC) distributions [3] (Fig. 2). This produced an estimate of the vocal tract filter of the target, $1/\tilde{A}_t(z)$. The original part of this speech conversion system resided in an additional mapping step. In order to increase the acoustic similarity between converted speech and target speech, we used a large database of target speech, and applied a units selection principle, similar to that used in unit selection for text-to-speech synthesis: we searched in the target database for a sequence of real target vocal tract filters $\{1/A_t(z)\}$ whose distance to the sequence of mapped filters $\{1/\tilde{A}_t(z)\}$ was minimized. The search for optimal target sequences was based on dynamic programming, for additionally optimizing the length of the real target filter sequences used (in order to avoid discontinuities when switching from one sequence to another). Converted speech was finally obtained by filtering some excitation signal with the sequence of real target vocal tract filters $\{1/A_t(z)\}$.

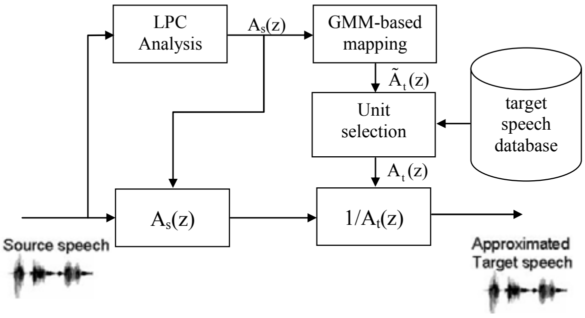


Fig. 2 Voice conversion in [1]

One of the main conclusions of [1] was that, if the *target speaker LP residual* was filtered by the sequence of

[1] http://www.enterface.net/enterface06/docs/results/databases/eNTERFACE06_ARCTIC.rar

mapped vocal tract filters $\{1/A_t(z)\}$, the converted speech sounded like "processed" target speech, and was therefore clearly identifiable as originating from the target speaker (but its quality was somehow discontinuous)[2]. In contrast, when the *source speaker LP residual* was used to drive the sequence of mapped vocal tract filters (as in Fig. 2), a lot of the source speaker identity was retained in the converted speech.

This led us to focus initially this year on source/tract separation, with separate mappings for source and tract parameters. The main initial idea was to use a recently developed source-tract model, the so-called AutoRegressive eXogeneous Lijlencrants-Fant model (ARX-LF) [4]. We also tested another mapping function (other than the one used in [1]), called STASC [2] and based on a simpler algorithm than [3] while still producing efficient vocal tract mapping.

A. ARX-LF

The source-filter model of speech production hypothesizes that a speech signal is formed by passing an excitation through a linear filter. In physiological terms, the excitation refers to the signal produced at the glottis and the filter represents the resonances caused by the cavities in the vocal tract. Linear prediction analysis is a basic source-filter model which assumes a periodic impulse train as the source signal for voiced sounds and white noise for unvoiced speech. Most voice conversion frameworks assume an LPC based analysis-synthesis approach, where only the LPC-based parameters are converted and excitations are left untouched. More elaborate parametric models of the excitation do exist and are interesting in terms of joint vocal tract and source conversion. In our experiments we have chosen to evaluate the LF model which models the voiced excitation by approximating the glottal flow derivative (GFD) with three parameters. Figure 3 illustrates a GFD waveform generated by the LF model. The three parameters of interest are open quotient, asymmetry coefficient and closed phase quotient.

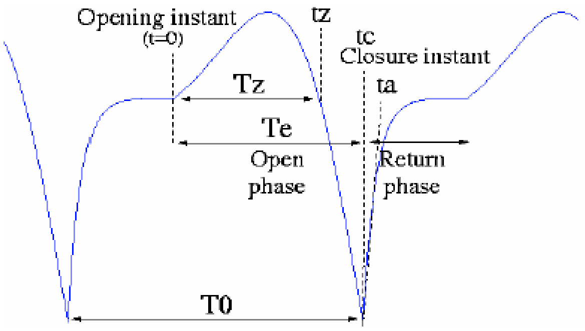


Fig. 3 Glottal flow derivative produced by the LF model

With the adoption of LF glottal pulse, the speech signal can then be expressed by an ARX (auto regressive exogenous) process [4]:

$$s(n) = \sum_{k=1}^{p} a(k)s(n-k) + u(n) + e(n) \qquad (1)$$

where *u* is the LF waveform and *e* is the residual noise and a(k) are the coefficients of the p^{th} order filter representing the vocal tract. Once an LF waveform is found for a given speech frame, deriving the filter coefficients is a trivial task. For ARX-LF analysis we have used an implementation based on the work of A. Moinet and N. D'Alessandro. Our implementation does not incorporate a residual noise factor, therefore *e*(n) is always 0 when synthesizing speech with our implementation of the ARX-LF model. The steps of the ARX-LF analysis can be summarized as follows:

1- Extract pitch at regular intervals (5ms) from the wav file.
2- Find the point of initial glottal closure instant (GCI) in each voiced segment in the utterance.
3- Use the first GCI as the anchor point to determine the remaining glottal closure instants in each voiced segment.
4- For each pitch period, search an LF derivative waveform codebook for the waveform which minimizes the error between actual speech frame and the ARX-LF synthesized frame.
5- Obtain the parameter set which produced this waveform (this is stored in the LF derivative waveform codebook, along with each waveform).
6- Given the LF parameter for each frame, determine the filter coefficients.

B. STASC

Speaker Transformation Algorithm using Segmental Codebooks (STASC) converts the voice of a source speaker to that of a target speaker maintaining high speech quality [2]. Figures 4 and 5 schematically depict training and conversion stages of STASC which are briefly described below.

(i) Training

First, alignment between the same sentences from source and target speaker is automatically performed by aligning Sentence HMMs for the (source, target) utterance pairs [2]. An advantage of this model is that it doesn't require any knowledge of the text or the language spoken. In every frame of source and target speech, acoustic feature vectors are extracted, including MFCCs, logarithm of energy and voicing probability – as well as their delta coefficients (18 features in total). For each source speaker utterance, the segmental *k*-means algorithm initializes an HMM whereas Baum-Welch algorithm is employed for training. Both source and target speaker utterances are force-aligned with

[2] Notice that, since the length of the source and target files were generally different, using target speaker excitation with modified source speaker vocal tract parameters implied to perform some alignment of the source and speaker files. This was achieved by applying Dynamic Time Warping (DTW) between source and target utterances.

this HMM using the Viterbi algorithm. A new state is added to the HMM topology every 40 ms of source speaker utterance. The mapping between the acoustic parameters of source and target speakers can subsequently be obtained based on this aligned data. For each HMM state, line spectral frequencies (LSF), fundamental frequencies (F_0), durations, energy and excitation parameters are computed. Their mean values over the corresponding source and target HMM states are stored in the source and target codebooks, respectively.

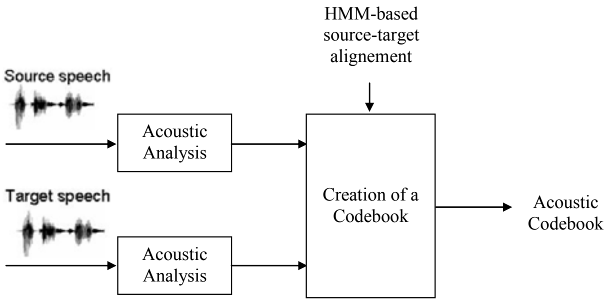


Fig. 4 Training stage flowchart of STASC

(ii) Conversion

Vocal tract and excitation characteristics are independently modified. Line spectral frequencies (LSF) are selected to represent vocal tract characteristics of each speaker since they are closely related to formant frequencies and, moreover, they can be reliably estimated. After pitch-synchronous linear prediction (LP) analysis of source speaker utterance, LP parameters are converted to LSFs.

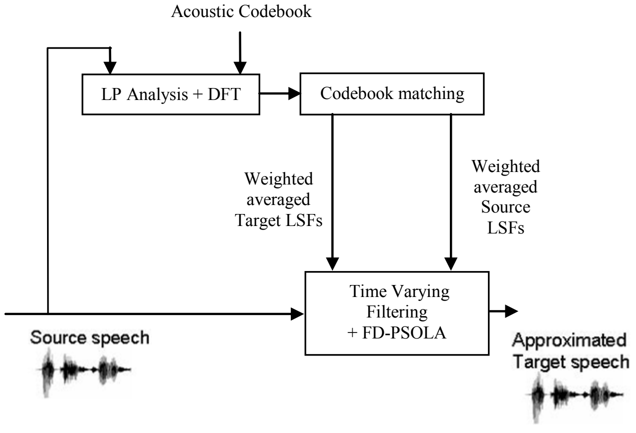

Acoustic Codebook
LP Analysis + DFT
Codebook matching
Weighted averaged Target LSFs
Weighted averaged Source LSFs
Time Varying Filtering + FD-PSOLA
Source speech
Approximated Target speech

Fig. 5 Conversion stage flowchart of STASC (after [2])

The (weighted) distance d_m between the LSF vector u of the input source frame and the m^{th} source codebook LSF vector is given by the following equations:

$$d_m = \sum_{n=1}^{P} k_n \left| u_n - C^s_{mn} \right| \qquad \text{for} \quad m = 1,\ldots,M \qquad (2)$$

$$k_n = \frac{1}{\arg\min\left(\left|u_n - u_{n-1}\right|, \left|u_n - u_{n+1}\right|\right)}, \quad n = 1,\ldots,P \qquad (3)$$

where m is the codebook entry index, M is the codebook size, n is the index of LSF vector entries, P is the dimension of LSF vectors (order of LP analysis), u_n the n^{th} entry of the LSF vector for the input source frame, C^s_{mn} is the n^{th} entry of the m^{th} source codebook LSF vector, and k_n are the LSF weights.

STASC further estimates the vocal tract characteristics of the target speaker, i.e. the n^{th} entry of the estimated target LSF vector $\bar{y}_n$:

$$\bar{y}_n = \sum_{m=1}^{M} \upsilon^m C^t_{mn} \qquad \text{for} \quad n = 1,\ldots,P \qquad (4)$$

where C^t_{mn} is the n^{th} entry of the m^{th} target codebook LSF vector, υ^m are the normalized codebook weights, and $\bar{\square}$ is used to show that □ is obtained through weighted average of codebook entries. The target LSF vector $\bar{y}_n$ is converted into target LP coefficients in order to obtain target vocal tract spectrum $\hat{H}^t(\varphi)$ (where φ is the angular frequency in radians. The frequency response $H^{VT}(\varphi)$ of the time-varying vocal tract filter for the current frame is then given by:

$$H^{VT}(\varphi) = \frac{\hat{H}^t(\varphi)}{H^s(\varphi)} \quad \text{or} \quad H^{VT}(\varphi) = \frac{\hat{H}^t(\varphi)}{\bar{H}^s(\varphi)} \qquad (5)$$

where the source vocal tract spectrum can be estimated using either the original or "estimated" LP coefficients (see [2] for details).

The frequency-domain pitch synchronous overlap-add algorithm (FD-PSOLA) is finally used for adapting the pitch of the source to the pitch range of the target.

III. Facial Expression Analysis, Mapping and Synthesis for Face Animation

Starting from video samples (face and shoulders) of the source speaker, and a photograph of the target speaker, we want to produce video samples of the target speaker acting and speaking as the source speaker.

Solving such a problem implies the following steps: (1) analyze the facial movements of the source speaker, using some parametric 3D facial model; (2) estimate the parameters of the same 3D model for the target speaker (3) normalize the facial movements of the source speaker relatively to some biometrical features of his face (4) animate the target 3D model by applying the same relative face movements as those measured on the source speaker.

While we had only considered steps 1, 3, and 4 in our previous work [1], by using an avatar that was already available (without having to build an avatar corresponding to an identified target speaker), this year's project browsed all 4 steps.

A. Face modelling

The parametric 3D facial model used in our project is the one defined in the MPEG-4 standard [5]. It is composed of Facial Definition Parameters (FDPs), which define the 3D position of a number of reference points on the face, Facial Animation Parameters (FAPs), which define frame-by-frame define movements of those reference points.[3]

B. Facial movement mapping

Face movement mapping is made easy by the fact that FAPs are actually not sent as absolute movements, but rather as movements normalized by the so-called Facial Animation Parameter Units (FAPUs), which can be obtained from the FDPs (they are basically some important biometric features, such as inter-ocular distance, mouth width, etc.). Applying measured face movements from source face to target face is thus as simple as imposing source FAPs on the 3D model of the target, and using the target FAPUs to interpret the source FAPs.

C. Face analysis

In [1], the analysis part mainly consisted in three tasks: detecting and tracking the face region, extracting facial features such as lips, nose, eyes and brows, and tracking these features throughout the source video.

The first part of analysis was done by computing an ellipse outlining the face. The ellipse parameters were estimated from an optical flow [8]. The centre of the ellipse, approximating the centre of the head, was used to track global head movements by assuming that the head centre moves around a sphere situated on the top of spinal cord. By projecting the displacement of the head onto this sphere, the angles of head rotation (pitch, yaw and roll angles) were approximately estimated.

The next task was to define and track the useful feature points for face components, which are lips, eyes, and eyebrows in this scenario. For this purpose, Active Appearance Model (AAM) approach [9] was used as a means of modeling and tracking the face components. Since AAMs require model training phase before they are used to process entire sequence of frames of a given video, a set of frames which cover a wide range of different facial expressions was selected and used to train the AAM. This training step requires manual labeling of feature points located around the desired face components over all frames in the training set. After the model was created by using the set of manually labeled points, feature tracking for face components could be performed easily. From the position of this feature points and global head movements, facial animations parameters were computed.

In this year's project, we focused on improving the global head movement tracking. In [1], global head movements and facial feature tracking were independent tasks. A way to improve the approximation of head rotation angles is to exploit features tracked from the AAM, especially rigid features such as eyes corners or point located at the beginning of the nose, between nostrils.

As we don't know the depth of these points it is not possible to directly compute rotation angles. In [10], the 2D image coordinates of feature points are mapped to 3D by assuming the projection is approximately scaled orthographic. Another solution is to compute an orthogonal projection of the centre of the head on the plane defined by the three points and compute relations between them. When the face is in a frontal position, the centre and its projection are one and the same in 2D.

Consequently, to compute relations between the eyes corners, the bottom of the nose and the head centre, we used a frontal head pose picture of the source. As the 3D rotation change angles and distances in 2D, we used ratios between features points to find points necessary to localize the centre.

Let *RE* be the corner of the right eye, *LE* the corner of the left eye, *N* the bottom of the nose and *C* the center of the head (Fig. 6). Let *RELE* be the straight line defined by *RE* and *LE*, *NRE* the straight line defined by *RE* and *N*, *REC* the straight line defined by *RE* and *C*, and *LEC* the straight line defined by *LE* and *C*. Let $I_{rec-nre}$ be the intersection between *REC* and *NRE*, and $I_{lec-nre}$ be the intersection between *LEC* and *NRE*. The following ratios are then computed:

$$R_1 = \frac{d(I_{rec-nre}, re)}{d(re, n)}, \quad R_2 = \frac{d(I_{lec-nre}, le)}{d(le, n)} \qquad (6)$$

where *d* is the Euclidian distance. For each frame, R_1 and R_2 are used to find the position of each intersection points on the corresponding segments and from the location of these points, the 2D coordinates of the head centre projection are computed.

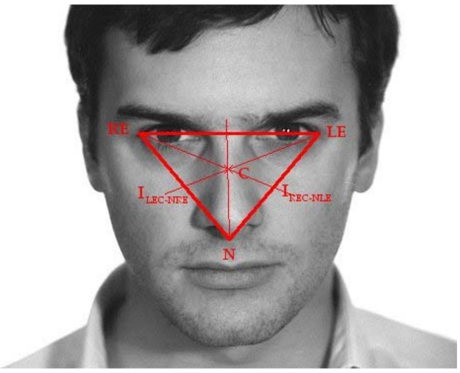


Fig. 6 Relations between the three feature points and the center of the head

To compute the depth of the centre and the rotation angles, we used the same method as in [1].

IV. Experiments and Results

In order to design the application depicted in Figure 1, we needed to choose a source and target speaker, make sure we

[3] Notice that how the movement of each reference point influences the final face rendering is not defined by MPEG4. Each face rendering software does it its own way.

could have access to a large amount of speech from the target (for the speech synthesis module), of a reasonable amount of aligned speech data for source *and* target (for the voice mapping module), and of some test speech and video data from the source (in order to test the complete system). The eNTERFACE06_ARCTIC database meets these requirements for the source speaker. It is composed of 199 sentences, spoken by one male speaker, and uniformly sampled from the CMU_ARCTIC database [2]. For each sentence, an .txt, a .avi, and a .wav file are available. The .avi file contains images with 320x240 pixels (Fig. 3), 30 frames per second, of the speaker pronouncing the sentence (Fs=44100 Hz). The .wav file contains the same sound recording as in the .avi file, but resampled to 16 kHz.

In the next paragraphs we expose the results we have obtained with the ARX-LF model and the STASC algorithm; we also report on an improvement we have made to the face analysis module of [1], and we show results related to modeling the target face, and to animating it.

A. ARX-LF

Copy synthesis experiments were carried out using the ARX-LF framework (Fig. 7). Here we discuss an informal evaluation of the stimuli.

(i) Copy-synthesis with fixed excitation

In this very simple copy synthesis experiment, a random LF parameter set was chosen from the codebook and the same parameter set was copied to produce all the voiced frames of an input utterance. The purpose of this experiment was twofold: to assess the quality of analysis-resynthesis when the source parameters are modified independently from the filter coefficients and to make an informal evaluation of how much difference in voice quality is perceived when different LF parameter sets are copied throughout an utterance.

The quality of the resynthesis with fixed excitation was acceptable and did not result in many artifacts. There were also perceived differences in voice quality between some parameter sets of the codebook, particularly regarding the breathiness and brightness of the voice. However, there were also many codebook entries which produced no perceptual difference when copied throughout an utterance. This led us to become more skeptical about the potential contribution of LF parameters to speaker identity.

(ii) Transplantation of target speaker features to source speaker

Since the results of last year's project indicated the existence of a lot of speaker specific information in the residual, our goal was to code some of that information with the LF parameters using the proposed framework. Therefore a natural experiment to conduct was to align the source and target frames in a parallel utterance and copy the relevant parameters of the target onto the source.

We have compared versions of utterances where only the filter coefficients were copied over, versus ones where both filter coefficients and LF parameters were copied over. We found that for our speaker conversion task, the speaker identity was still mostly coded in the filter coefficients and copying the LF parameters made no perceptual difference in terms of speaker identity. In fact, changing only the LF parameters of the source frames to match those of the target resulted in a stimulus which sounded very much like the source. This again clearly questions the correlation between of LF parameters and speaker identity

In addition, we have tried copy-synthesis experiments on an emotion conversion task using an expressive speech corpus from a female speaker. For neutral to angry copy-synthesis on stimuli, we also found that most of the harshness of anger was coded in the filter parameters rather than the source. On the other hand, going from neutral to sad speech, there was a positive contribution of the LF parameters to perception of sadness.

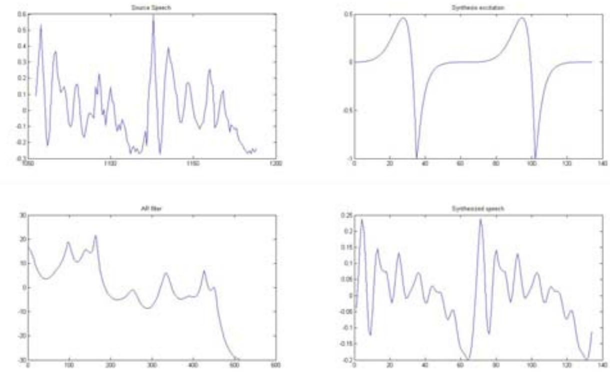


Fig. 7 An example of ARX-LF analysis-synthesis. Top-left: original speech frame; Top-right: Estimated Glottal derivative waveform; Bottom-left: Frequency response of the estimated vocal tract filter; Bottom-right: resulting synthetic speech frame

There are possible reasons why the ARX-LF framework may not have helped as well as we initially hope. We list them here, for future work:

- Glottal waveforms do not contain as much voice quality information as expected.
- It is possible that the current parametric framework is not adapted to model important voice quality information such as spectral tilt. (A suggestion here is to apply pre-emphasis to the speech signal during analysis so as to flatten the vocal tract spectrum and force the modeling of spectral tilt in the LF parameters.)
- Adding a component modeling residual noise may help. Probably this component is not even *really* noise-like.
- A better automatic GCI detection algorithm is crucial for this implementation but for the experiments performed here, the GCIs were manually corrected for best analysis/resynthesis results.

– The codebook of LF parameters may be expanded to contain more extreme parameter values and with higher resolution.

B. STASC

We reimplemented the STASC voice conversion system described previously. Although the implemented system is very similar to the one described in [2], some differences exist. We had at our disposal a sentence HMM alignment that has been produced as described in [2][4]. Instead of using the average of HMM state features as elements of the mapping codebook, we converted the alignment into a frame-to-frame alignment, leading to a frame-based mapping codebook of LSF parameters, *F0* and energy. We then cleaned this codebook by using the spectral distance confidence measure, the *F0* distance confidence measure and the energy distance confidence measure described in [2]. The original codebook counts 81870 pairs of frames while the cleaned codebook contains 69076 pairs. The frame length is 30 ms, with a shift of 10 ms. The second difference occurs in the conversion step, where we do not compute AR coefficients on pitch synchronous frames, but also on 30ms length frames, shifted every 10ms. Two types of tests have been made, referenced below as *method a* and *method b*.

Method *a* is depicted in Fig 8. LSF coefficients are computed for each frame of the source utterance. The *N* nearest codebook entries are selected, and a weighted sum of corresponding target LSF coefficients is computed. LSF coefficients of the source and of the target allow, after conversion into AR coefficients, to build the corrective filter as the division of the vocal tract frequency response of the target by that of the source. Each source frame is then filtered by the corrective filter. Speech is resynthesized by overlap-adding the resulting voice-converted frames.

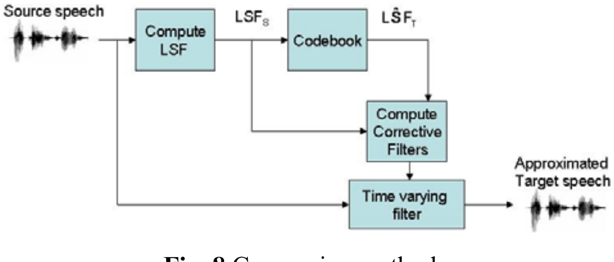


Fig. 8 Conversion method *a*

In order to better check the efficiency of the vocal tract conversion, another synthesis method has been implemented. This method cannot be seen as a genuine conversion system, as it uses the target utterances, but its results give an idea on how well the vocal tract (alone) is transformed from one speaker to another. It uses the target LP residual, and filters it with LSF codebook outputs converted into AR coefficients. The system is represented in Fig 9.

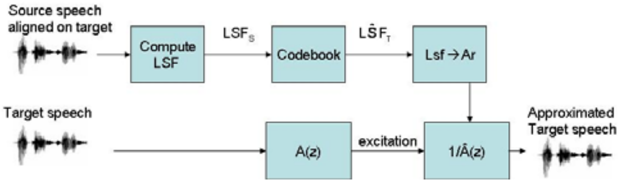


Fig. 9 Conversion method *b*

In theory, if the codebook-based mapping of the valv tract from source to target was perfect, the approximated target speech should be identical to the target speech.

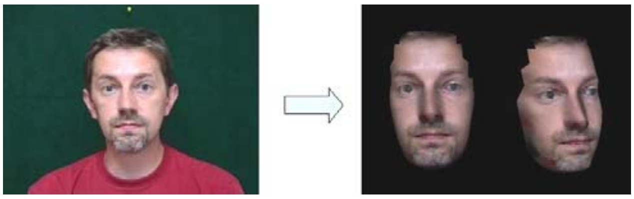
Notice that, as we use the source frames to find the approximated LSF coefficients of target frames and then use the target wavefile for synthesis, source frames and target frames need to be aligned. This alignment is done by dynamic time warping, using the implementation provided by Dan Ellis[5].

Method *a*, the real conversion test, leads to a very good quality, but the identity change is incomplete: it moves from source to something between source and target voices, but still closer to the source. A pitch modification of 10 percent on the converted speech improves a bit the similarity with the target speaker. This pitch modification was done using Praat[6] software.

Method *b* lead to a much better similarity with target speaker (but again, we are using the target excitation, so tests *a* and *b*cannot be compared) although the quality of the converted speech is degraded. Actually, the quality and similarity of the converted voice is somehow similar to the ones obtained from eNTERFACE06 project 4 [1], but the algorithm used this year is very much simpler.

C. Face modeling

One of the tools we used for face modelling was the PhotoFit software. This software produces a 3D head model of a person from the person's photograph, as shown in figures 10 and 11. It also needs an FDP file that corresponds to this photograph. FDP file can be created using the visage|annotator software. It displays input 2D picture and automatically detects and marks the feature points in the face. After that some points can be manually corrected. Positions of these points correspond to the positions of the MPEG-4 facial Feature Points (FPs) [7].



[4] They have been added to the in the eNTERFACE06_ARCTIC archive.

[5] http://labrosa.ee.columbia.edu/matlab/dtw/

[6] http://www.fon.hum.uva.nl/praat/

Fig. 10 Input photograph of the source speaker and generated 3D head model by PhotoFit

PhotoFit basically takes a generic 3D head model and deforms it in such a way that labelled feature points fit to their positions taken from the FDP file. Other points of the model are also moved so that they form a smooth surface of the face with labelled feature points.

Generation of the facial skin textures from uncalibrated input photographs as well as the creation of individual textures for facial components such as eyes or teeth is addressed in [6]. Apart from an initial feature point selection for the skin texturing, these methods work fully automatically without any user interaction. The resulting textures show a high quality and are suitable for both photo-realistic and real-time facial animation.

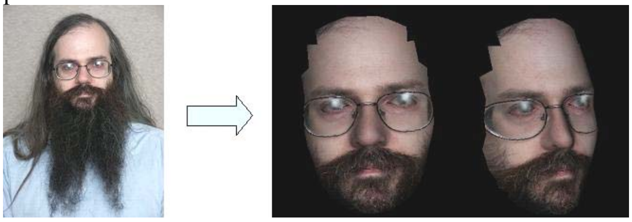


Fig. 11 Input photograph of the target speaker and generated 3D head model by PhotoFit

The 3D input mesh is parameterized over the 2D domain (0, 1) x (0,1) (part of $\mathbb{R}^2$) in order to obtain a single texture map for the whole mesh. In [6], the face mesh is topologically equivalent to a part of a plane, since it has a boundary around the neck and does not contain any handles. The face mesh can be "flatten" to a part of a plane that is bounded by its boundary curve around the neck. PhotoFit uses described methods to create facial skin textures, but it parameterizes a mesh with a cube instead of a disk.

It is important to mention that the 2D input picture should be a frontal photograph of the person, and should contain the person's face and shoulders. Also, the face should be in the neutral position according to the MPEG-4 FBA specification. If the person on the picture has an expression it will keep that expression through the whole animation, i.e. if the person is smiling generated model will always be smiling.

D. *Face analysis*

In [1], each frame in the training set had been manually labeled with 72 landmarks by using the AAM-API software [10] (freely available for non-commercial use such as research and education). In order to obtain an accurate mapping from the 2D pixel positions to the MPEG-4 parameters, the annotation of the images in the training set should closely match the MPEG-4 FDPs.

The global head movements and the feature points tracking were done on 10 videos. To compute the head ellipse, OpenCv Lib (available on the Intel website) has been used. The calculated values for animation has been smooth, since the measurements in the tracking process are noisy and small scale differences in the parameters for the simulation process may have large effects in the resulting animation. A Kalman Filter was used for this purpose with a state model consisting of positions and velocities of all the key feature points on the face. Fig. 12 shows the results of the global head movements tracking. On the whole, we have noticed an improvement in the head center tracking. The new method is also more robust and has allowed to eradicate big errors due to the use of the optical flow (Fig. 13).

E. *Face synthesis*

We created 10 animations using XFace interface which is an MPEG-4 based open source toolkit for 3D facial animation, developed by Balci [11]. This is a straightforward process: once the FAP values have been computed, the XFace editor (or any other suitable player) can directly synthesize them (given that the parameters file format is correct). Fig. 14 shows the results of the generation of an animation with Xface. The animation is not only depending of the computation of the FAPS, and consequently of the quality of the tracking, but also of the 3D model.

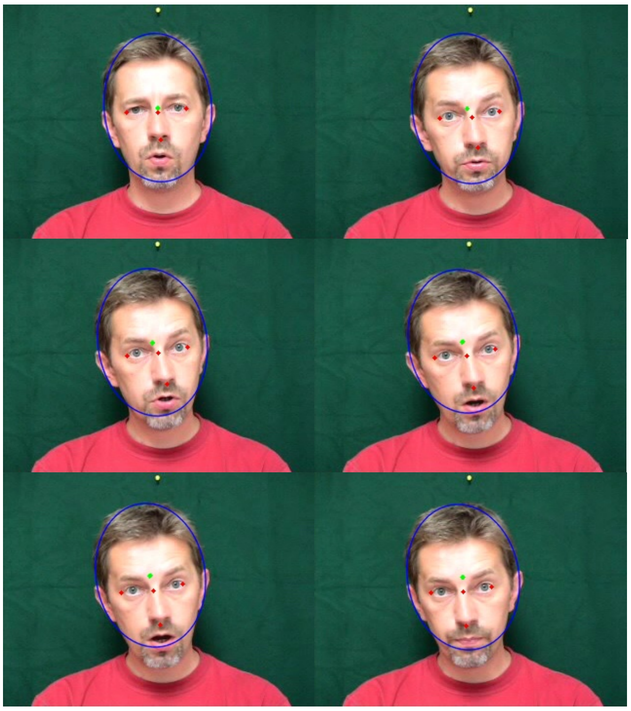


Fig. 12 Results of the global head movements tracking .The ellipse and the green point are the results of last year, red points is the new tracking method.

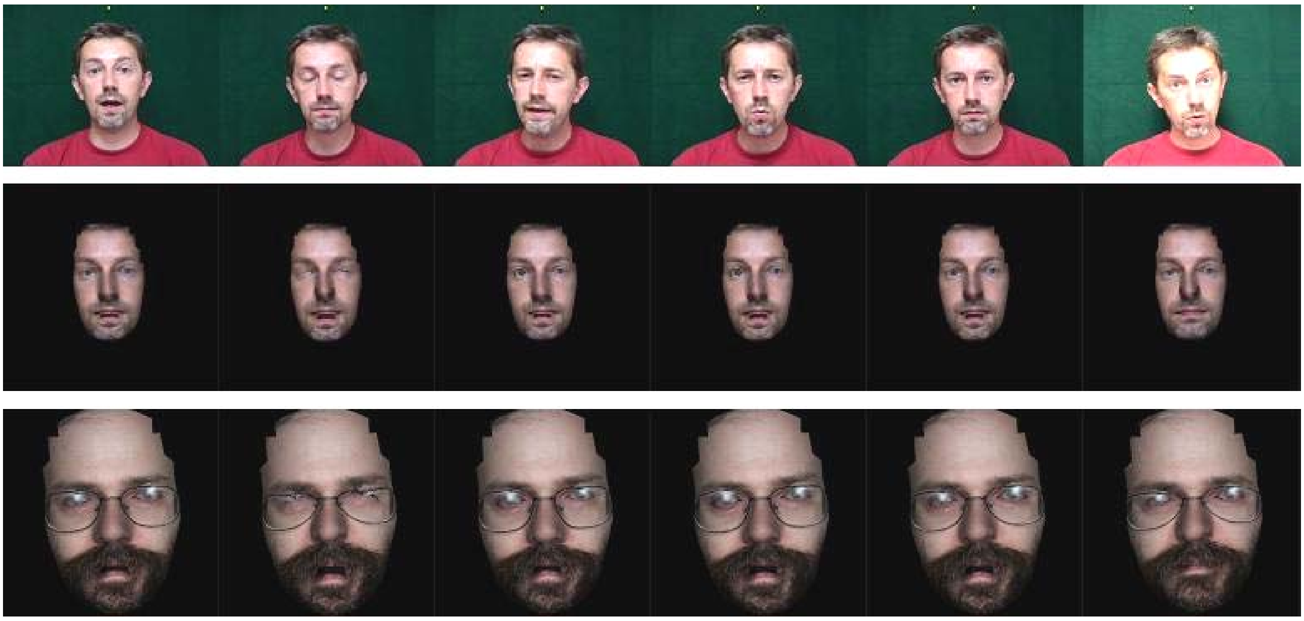


Fig. 15 (Top) Frames extracted from a video of the source speaker; (Center) Frames obtained by animating a 3D model of the source; (Bottom) Frames obtained by animating a 3D model of the target.

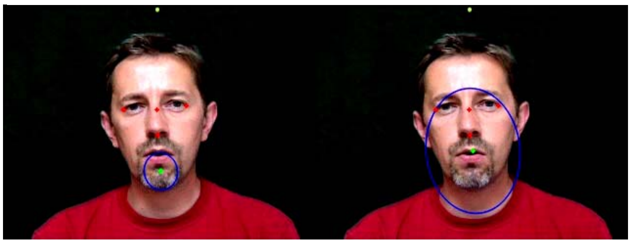


Fig. 13 Errors occurring with the ellipse computed from the optical flow.

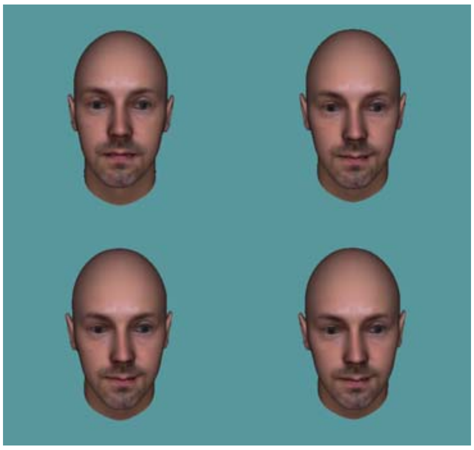


Fig. 14 Animation created with XFace

F. Face synthesis

PhotoFit uses commercial software visage|SDK [9] for animation and rendering of the generated model. visage|SDK is a Software Development Kit for MPEG-4 character animation. It includes the following main capabilities:

- Animating virtual characters using MPEG-4 Face and Body Animation Parameters (FAPs and BAPs).
- Real-time or off-line character animation driven by SAPI-5 speech synthesis, with on-the-fly rendering of the virtual character and MPEG-4 FBA bitstream output.
- Real-time or off-line lip sync from audio file or microphone, with on-the-fly rendering and MPEG-4 FBA output.
- Interfaces for plugging-in own interactive or off-line animation sources and controls.
- Coding, decoding, merging and other operations on MPEG-4 FBA (Face and Body Animation) bitstreams.

The analysis tool built during eNTERFACE'06 produces ASCII FAP files, and visage|SDK reads binary FBA files. To connect these two tools we had to write code that reads FAP files and calls visage|SDK functions for applying read values of FAPs to the generated model. Animated models of source and target speaker models are shown in figure 15.

V. Conclusions

In this paper we describe a multimodal speaker conversion system, based on the simultaneous use of facial animation detection, 3D talking face synthesis, and voice conversion.

We first try to take advantage of a recently developed source-filter estimation algorithm, namely the ARX-LF model, to perform parallel conversion of voice source parameters and of vocal tract parameters. Copy synthesis using ARX-LF gives acceptable results (although the resulting quality is very sensitive to GCI detection stability), but transplanting target parameters into a source utterance leads to very irregular speech quality.

We then test L. Arslan's STASC algorithm, in a simplified implementation. The results are much more stable, although the ID conversion is still incomplete.

Face conversion is based, as initiated in [1], on the MPEG4 FPSs, FAPS, and FAPUs. After modeling the source and the target speaker faces with the PhotoFit software, we drive the speaker 3D face model using the FAPs of the source. We have also improved the face tracking algorithm, by computing the global head position from the postions of the eyes and nose rather than by simplifying the face shape into an ellipsis.

The results we have obtained so far are more complete then those obtained in [1], although there is still much room for improvement.

The face tracking algorithm still provides only an approximation of the source speaker movements. The face rendering systems we have tested do not prevent the synthetic face from performing impossible facial movements. Last but not least, the voice conversion algorithms still provides better ID conversion if we keep the target LP residual untouched, which seems to shows that the LP residual still contains some of the speaker identity (although the opinion of all team members did not converge on this last conclusion).

Acknowledgment

We are grateful to Prof. F. Marques, who collaborated to the initial steps in the project design, but could finally not attend eNTERFACE'07.

We thank A. Moinet and N. D'Alesandro for providing access to their ARX-LF Matlab code.

We also want to thank Alan Black, who provided ID-photos and accepted that we make a demo avatar of him.

References

[1] Dutoit, T., Holzapfel, A., Jottrand, M., Marquès, F., Moinet, A., Ofli, F., Stylianou, Y., "Multimodal Speaker Conversion — his master's voice. . . and face —", *Proc. eNTERFACE'06 workshop*, Dubrovnik, 2006, pp. 34–45.

[2] Turk O., Arslan L., "Robust Processing Techniques for Voice Conversion", Computer Speech and Language, 20 (2006) 441–467.

[3] Y. Stylianou, O. Cappé, and E. Moulines, "Continuous probabilistic transform for voice conversion," *IEEE Trans. on Speech and Audio Processing*, vol. 6, no. 2, pp. 131–142, 1998.

[4] D. Vincent, O. Rosec, T. Chonavel, "A New Method for Speech Synthesis and Transformation Based On an ARX-LF Source-Filter Decomposition and HNM Modeling", Proc. ICASSP 2007, IV 525-528.

[5] I.S. Pandžić and R. Forchheimer, "MPEG-4 Facial Animation - The standard, implementations and applications", John Wiley & Sons, 2002.

[6] M. Tarini, H. Yamauchi, J. Haber and H. P. Seidel, "Texturing Faces". Available: http://vcg.isti.cnr.it/~tarini/pap0A/GI2002.pdf.

[7] Visage Technologies, "Visage|SDK", http://www.visagetechnologies.com/products_sdk.html

[8] M. E. Sargin, F. Ofli, Y. Yasinnik, O. Aran, A. Karpov, S. Wilson, E. Erzin, Y. Yemez, , and A. M. Tekalp, "Combined gesture-speechanalysis and synthesis," in *Proc. of the eNTERFACE'05 workshop*, Mons, Belgium, August 2005.

[9] T. F. Cootes, G. J. Edwards, , and C. J. Taylor, "Active appearance models," in *Proc. European Conf. on Computer Vision*, vol. 2, 1998, pp.484–498.

[10] H. Tao, R. Lopez, T. Huang, "Tracking Facial Features using probabilistic networks", *Proceedings of the 3rd International Conference on Face and Gesture Recognition*, 1998, p. 166-171

[11] K. Balci, "Xface: Mpeg-4 based open source toolkit for 3d facial animation," in *Proc. Advance Visual Interfaces*, 2004, pp. 399 402.

Levent Arslan has graduated from Bogazici University, Turkey in 1991. He received his M.S. and Ph.D. degrees from Duke University, USA in 1993 and 1996 respectively. He has worked at Texas Instruments and Entropic Research until 1998. Since then he has been teaching at Bogazici University. His main research interests are Voice conversion, speech recognition, speech synthesis, and Speech enhancement. He has published 13 journal papers and 70 conference papers in speech processing. He holds 9 international patents.

Thierry Dutoit graduated as an electrical engineer and Ph.D. in 1988 and 1993 from the Faculté Polytechnique de Mons, Belgium, where he is now a professor. He spent 16 months as a consultant for AT&T Labs Research in Murray Hill and Florham Park, NJ, from July, 1996 to September, 1998. He is the author of two books on speech processing and text-to-speech synthesis, of a forthcoming book on signal processing, and the coordinator of the MBROLA project for free multilingual speech synthesis. T. Dutoit was an Associate Editor of the IEEE Transactions on Speech and Audio Processing (2004-2006) and is a member of the INTERSPEECH'07 organization committee. He was the initiator of eNTERFACE workshops and the organizer of eNTERFACE'05.

Zeynep Inanoglu graduated with an Electrical Engineering degree from Harvard University in 1999. Between 1999 and 2002 she worked as a software engineer and product manager at Angel.com, focusing on agile development and dialog systems. Subsequently she received her masters degree in Speech and Language Processing at the University of Cambridge and is currently in her final year of her PhD in Cambridge, focusing on emotional speech synthesis and prosody modeling.

Matthieu Jottrand holds an Electrical Engineering degree from the Facult´e Polytechnique de Mons since June 2005. He did his master's thesis in the Image Coding Group of Link¨oping Institute of Technology. Matthieu is a researcher fromTCTS lab (FPMs) since September 2005. He is currently working in the field of ASR for the IRMA project (development of a multimodal interface to search into indexed audiovisual documents) and just started a PhD thesis under the supervision of Thierry Dutoit.

Maria Markaki holds a degree and a MSc in Physics from Athens University, and a MSc in Computer Science from University of Crete. From 1995 until 2001 she was a researcher in the Institute of Applied and Computational Mathematics (IACM - FORTH). She is currently working in the field of audio indexing for the SIMILAR project and pursues a PhD thesis under the supervision of Yannis Stylianou.

Igor S. Panzic is an Associate Professor at the Department of Telecommunications, Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia. His main research interests are in the field of computer graphics and virtual environments, with particular focus on facial animation, embodied conversational agents, and their applications in networked and mobile environments. Igor also worked on networked collaborative virtual environments, computer generated film production and parallel computing. He published four books and around 60 papers on these topics. Formerly he worked MIRALab, University of Geneva, where he finished his PhD in 1998. He was a visiting scientist at AT&T Labs, at the University of Linköping and at Kyoto University. Igor was one of the key contributors to the Facial Animation specification in the MPEG-4 International Standard for which he received an ISO Certificate of Appreciation in 2000.

Murat Saraçlar received his B.S. degree from Bilkent University, Ankara, Turkey in 1994. He earned both his M.S.E. and Ph.D. degrees from Johns Hopkins University, Baltimore, MD, USA in 1997 and 2001 respectively. He worked on automatic speech recognition for multimedia analysis systems from 2000 to 2005 at the AT&T Labs—Research. In 2005, he joined the Department of Electrical and Electronic Engineering at Bogazici University as an assistant professor. His main research interests include all aspects of speech recognition, its applications, as well as related fields such as speech and language processing, human-computer interaction and machine learning. He authored and co-authored more than two dozen papers in refereed journals and conference proceedings. He has filed four patents, both internationally and in the US. He has served as a reviewer and program committee member for various speech and language processing conferences and all the major speech processing journals. Dr. Saraçlar is currently a member of ISCA and IEEE, serving as an elected member of the IEEE Signal Processing Society Speech and Language Technical Committee (2007-2009).

Kristina Stanković just finished her undergraduate studies at Faculty of Electrical Engineering and Computing, the University of Zagreb, Croatia, and she will also start her PhD studies there. She is a member of Human-Oriented Technologies Laboratory (HOTlab). Her main interest is in the field of face and body animation, and its application with Internet and mobile technologies.

Yannis Stylianou is Associate Professor at University of Crete, Department of Computer Science. He received the Diploma of Electrical Engineering from NTUA, Athens, in 1991 and the M.Sc. and Ph.D. degrees in Signal Processing from ENST, Paris, France in 1992 and 1996, respectively. From 1996 until 2001 he was with AT&T Labs Research (Murray Hill and Florham Park, NJ, USA) as a Senior Technical Staff Member. In 2001 he joined Bell-Labs Lucent Technologies, in Murray Hill, NJ, USA. Since 2002 he is with the Computer Science Department at the University of Crete. He was Associate Editor for the IEEE Signal Processing Letters from 1999 until 2002. He is Associate Editor of the EURASIP Journal on Speech, Audio and Music Processing. He served on the Management Committee for the COST Action 277: "Nonlinear Speech Processing" and he is one of the two proponents for a new COST Action on Voice Quality Assessment.

Aurélie Zara received her Master Degree in computer sciences from Université Paris XI, Orsay, France, since 2006. She did her master thesis on the modelisation of multimodal emotional interaction between users and virtual agents at LIMSI-CNRS. She is currently doing a Phd thesis at France Télécom Orange Labs. Her research is on the impact of multimodal expressions of emotions of a virtual agent interacting with an human.