An MPEG-4 Facial Animation System Driven by Synthetic Speech

Claudio Lande and Gianluca Francini
CSELT - Centro Studi e Laboratori Telecomunicazioni S.p.A.
Via Reiss Romoli, 274, 10148 Torino (Italy).
Claudio. Lande@cselt. it, Gianluca. Francini@cselt. it

Abstract

This paper' provides an overview of the activities that have led to the development of a prototype application of a "talking head" compliant with MPEG-4 specification. The work done so far fits in the context of one of the recent work-items defined by the ISO/IEC JTC1/SC29/WG11 world-wide known as MPEG. This IS0 Working Group is now in the process of defining a unifying framework where natural and synthetic audio-visual objects can be combined and rendered as a unique bouquet of synchronised interactive media.

The developed facial animation system implements the following features: animation of predefined / downloaded face models, animation of face models driven either by speech synthesis applications or by MPEG-4 animation parameters. To improve photo-realism, face models can be texture mapped and calibrated according to the countenances of real people.

1. Introduction

The current MPEG standardisation phase (MPEG-4) is aimed at continuing the development of compression techniques (as in MPEG-1, MPEG-2) but at the same time aims at studying and defining innovative technologies that allow:

- interaction with audio-visual contents populating a scene
- composition of audio-visual material of different nature (i.e. synthetic and natural)

In the context of these last activities [1], [2], [3], identified with the acronym "SNHC" - Synthetic and Natural Hybrid Coding -, it is of special interest the activity of the SNHC Face and Body Animation (FBA) working group that is focused on:

- delineation of parameters for face (and body) animation/definition
- integration of Text To Speech (TTS) synthesis functionality

Face animation is based on the development of the following sets of parameters: Facial Animation Parameters (FAPs) and Facial Definition Parameters (FDPs). FAPs are defined in a neutral way with respect to face models. This feature allows having a single set of parameters that can be used regardless of the face model used by the application. Most FAPs describe atomic movements of the facial features (e.g. open_jaw, depress-chin, raise-nose, head roll, etc.). FAPs combinations give place to all the possible facial expressions. If a continuous animation must be performed, a sequence of FAPs must be applied to the face model: for each time instant (frame of the sequence) a set of FAPs must be used to update the model.

FDPs are used either to adapt a predefined model to a particular face *(calibration)* or to download a new model.

2. Architecture of the Facial Animation System

The general architecture of the system, borrowed from MPEG-4, is shown in Figure 2-1, s composed by a communication front-end that receives a composite stream and splits it in the different elementary streams (ES). Each ES feeds the associated decoder that, in turn, generates a stream of decoded data that are sent to the animation and synthesis subsystems. The data read from the ES can be:

- FAPs for model animation,
- FDPs for model calibration/downloading,
- Text that is converted by TTS into speech and into the corresponding phonemes.

Of course, in a full compliant MPEG-4 viewer, there could be a larger set of decoders for video, audio, 2D/3D graphics, still images, text, etc.

[1] The work described in this paper is supported by the European Commission via the ACTS EMPHASIS and PANORAMA projects

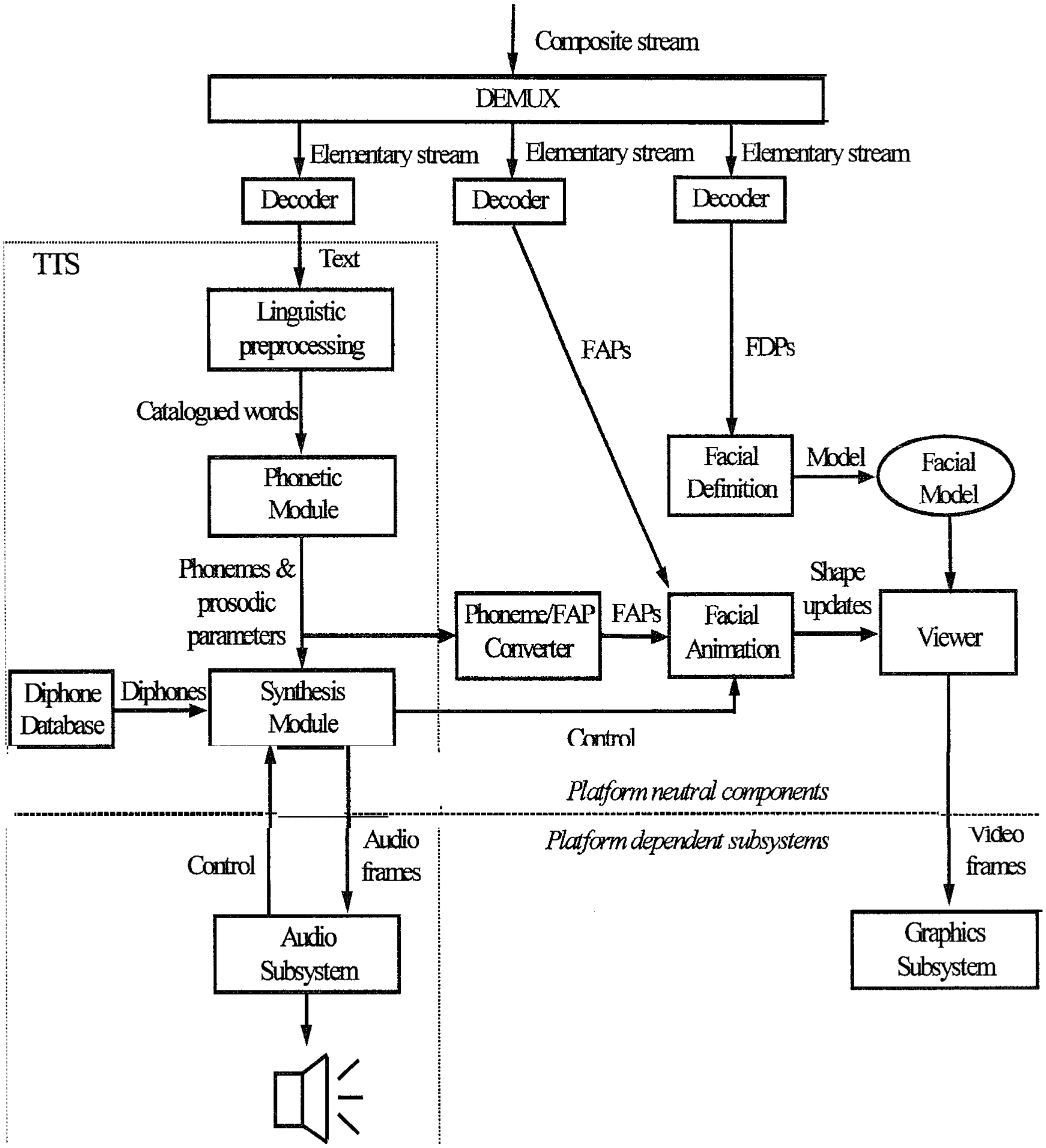

Figure 2-1. Reference architecture

The interface between TTS and face animation module provides:

- a function for reading phonemes (used to drive the face model) and their duration
- a control mechanism for the synchronisation with audio subsystem

The animation subsystem performs the animation and the visualisation of a face model and takes care of the mutual audio-visual synchronisation.

3. Calibration of the Face Model

In order to improve the look of the predefmed face model, a calibration system has been developed that produces a textured model with the countenances of a given person, on the basis of a front picture of the person itself. Using only a front view, no information are available about the depth of the face, hence the obtained calibrated model will not match exactly the features of the person. For example, the model nose will be always straight even if the subject has a snub or aquiline nose.

The calibration algorithm requires neither a priori knowledge of the camera parameters nor the relative position of the person and the camera (for details about the formulae, see [4]).

The first step is to manually determine in the image the position of a subset (22) of the MPEG-4 feature points, as depicted in Figure 3- 1 (right).

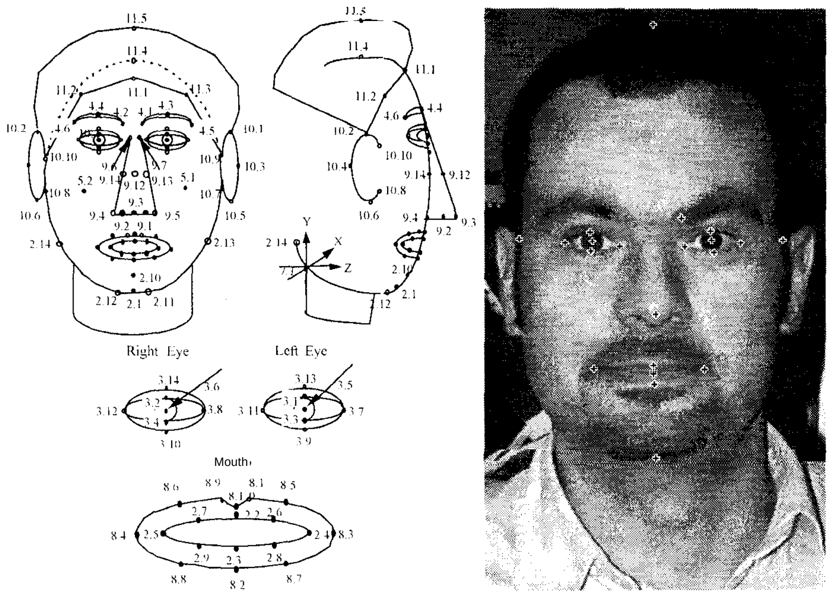

Figure 3-1. (Left) MPEG-4 feature points • (Right) The set of used points

The calibration algorithm modifies a wire-frame template so that, at the end of the process, the projection of the feature points of the calibrated model coincide with the marked feature points in the photo. In order to do this, a projection with the focus in the origin is performed (Figure 3-2).

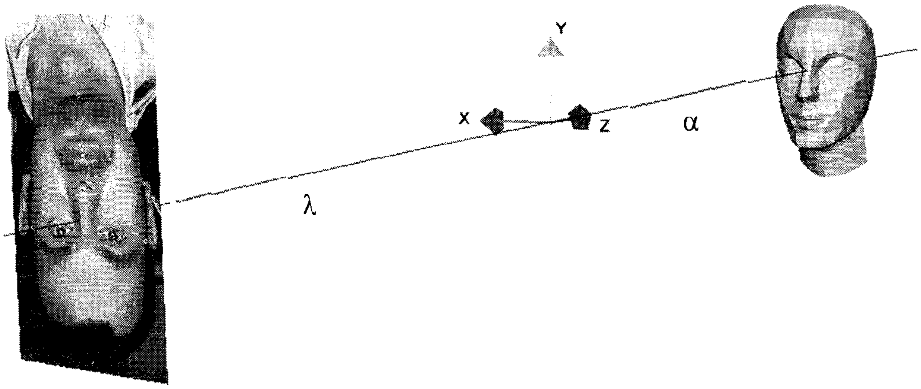

Figure 3-2. Projection with the focus in the origin

The template is positioned at a fixed distance a from the origin of the axes. This distance is arbitrary and represents a mean distance. The picture is centred with the Z axes considering the positions of the point 11.5 top of head, 2.1 tip of chin, 10.9 and 10.10 tip of cheekbones / ears. The distance λ between the image and the origin is the distance in which the projection of point 10.9 coincides with the corresponding 2D point in the photo. Let ($x_{10.9}$,

$y_{10.9}$, $z_{10.9}$) be the co-ordinates in the template of the point 10.9 (after the positioning at a distance along the *Z* axes) and ($X_{10.9}$, $Y_{10.9}$) the co-ordinates of the point in the picture then λ is obtained as:

$$\lambda = -\frac{x_{10.9} \cdot Z_{10.9}}{X_{10.9}}$$

After this operation, the wire-frame is translated and scaled along the Y axes in such a way that the projection of the points 11.4 and 2.1 coincides with the analogous points in the image. Let *(X, Y, Z)* the old co-ordinates of each point of the wire-frame, the new co-ordinates *(X', Y', Z')* are obtained as:

$$\begin{vmatrix} X' \\ Y' \\ Z' \\ 1 \end{vmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{y_{2.1} \cdot Z_{2.1}}{\lambda \cdot Y_{2.1}} & 0 & -\frac{Z_{11.4}Y_{2.1} + Z_{2.1}Y_{11.4}}{Z_{11.4} + Z_{2.1}} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Now, the projection of the new wire-frame matches the area occupied by the face in the photo. Given that different persons have different anthropometric measures, it is necessary to apply a general transformation on the mesh in order to recreate the same ratio between the height of the forehead and the height of the whole wire-frame. For this transform, the MPEG-4 point 4.1 (end of left eyebrow) is considered. The Y co-ordinates are altered so that the Y 11.4 and Y2.1 co-ordinates remain unchanged and the projection of the Y co-ordinate of point 4.1 matches with the corresponding position marked on the image. The transform is represented by the

The transform affects all vertices except those belonging to the eyes (pupils, iris, sclera and eyelids). Those points are subsequently translated centring the pupils' positions of the images. After this operation, the position of the vertices of the eyelids is changed with an affme transform in order to adapt the standard ellipse of the eyes with the shape of the person's eyes. The next phase consists in the scaling of the nose region, so that the

parabolic function passing in the three points ($Y_{1\ 1.4}$, $Y_{1\ 1.4}$), ($Y'_{4.1}$, $Y_{4.1}$), ($Y_{2.1}$, $Y_{2.1}$), where $Y'_{4.1}$ is the intersection of the plane Z = 24.1 and the projection of the point $y_{4.1}$ from the image in the space.

$$Y'_{4.1} = -\frac{Y4.1 \cdot Z_{4.1}}{\lambda}$$

Figure 3-3 shows the models before and after the transform.

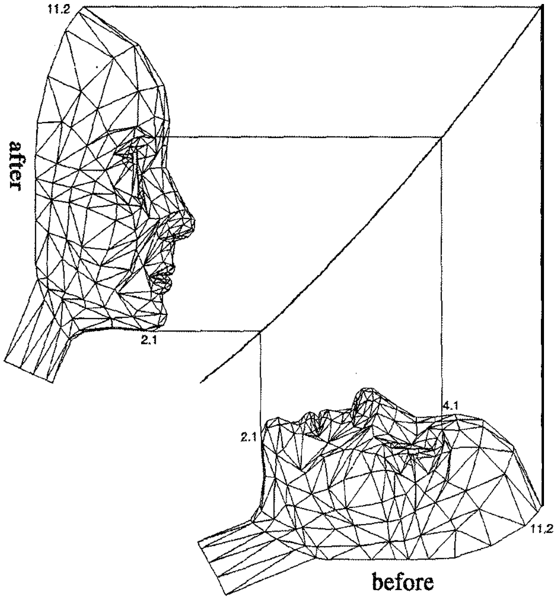

Figure 3-3. Global transform using a parabolic function

projection of the nose tip (9.3) matches the corresponding point in the image.

The final operation consists in changing X and Y co-ordinates of the triangles below the nose, (mouth and chin) with an afiine transform using the marked position of the eight points 2.2, 2.4, 2.3, 2.5, 8.1, 8.3, 8.2 and 8.4. The complete process is shown in Figure 3-4.

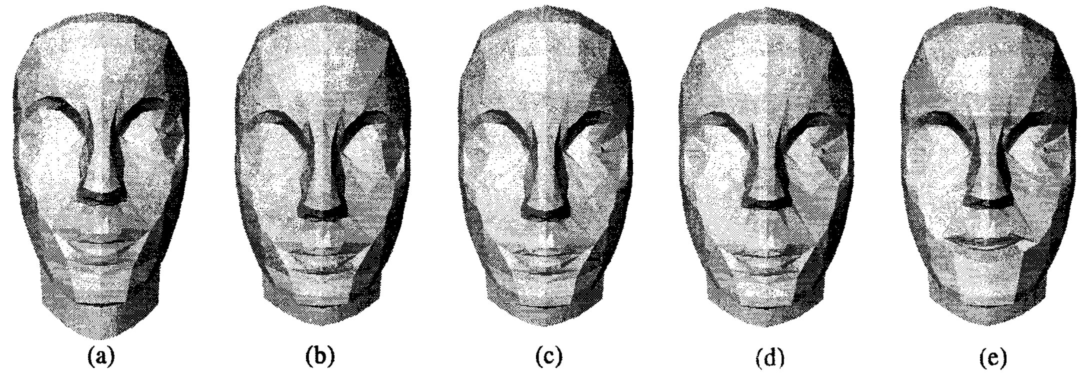

Figure 3-4. (a) Y scaling, (b) Global transform, (c) Eyes repositioning, (d) Scaling of the nose (e) The final model, after the affine transform of the mouth's region.

The final operation consists in the determination of the texture binding co-ordinates for each vertex, which is obtained simply projecting the vertices in the image plane. After that, the model is ready for texture mapping; in Figure 3-5, two views of the textured model are shown.

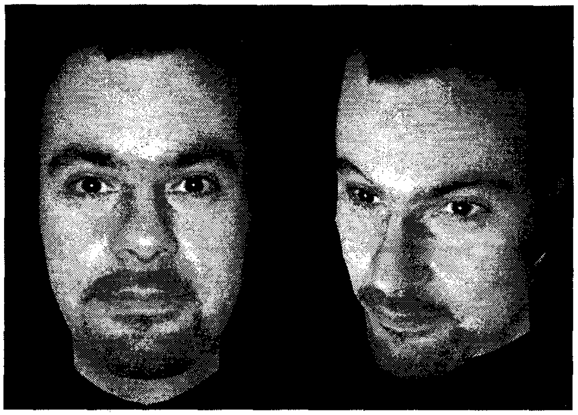

Figure 3-5 Two views of a calibrated and texture mapped synthetic model

3.1 Evaluation of the Calibration Technique

The described calibration technique, that is original of this work, has proved efficient in terms of execution time and satisfactory in terms of quality of results. The calibrated face models match very well the somatic features of the target persons so that the animation of these models does not show any unpleasant defect.

Matching synthetic model and target human face feature points is especially important for highly moving parts like mouth and eyes in order to have good-looking animations.

The reason why only 2D feature points (extracted from a font picture of a given individual) have been used, is that it is rather difficult to extract 3D feature points from human faces unless expensive hardware like cyber-scan equipment is used. This approach allows everybody to create his or her 3D-face model starting from a simple front picture. Further developments in face calibration include using 3D feature points defined by MPEG-4 and completing the face model to include ears, tongue and the back of the head.

4. Animation of the Face Model

The face model is animated by 59 FAPs that are relative to eyes, eyelids, eyebrows, lips, jaw, nose, cheeks and head rotations.

The face model is made as a mesh of polygons; the application of MPEG-4 Facial Animation Parameters (FAPs) on the model involves the displacement of a given set of vertices of the mesh with respect to the face neutral position.

If FAP *i* at time *t* (*t* can be considered the number of the frame) has intensity x,(t) ($x_i(t)$ is a three elements vector that indicates the direction along which the feature point associated with FAP *i* moves) and it affects vertices v_k., then the formula for the application of FAP *i* is:

$$v_k(t) = v_k(0) + FAPU_i * \alpha_{ki} * x_i(t) \qquad (1)$$

where $v_k(0)$ is the position of vertex v_k in the neutral position, *FAPU*, indicates the Facial Animation Parameter Unit used for FAP *i* (measurement units depend on FAPs), α_{ki} is a coefficient that expresses how much vertex v_k is affected by FAP *i* ($0 < \alpha_{ki} \leq 1$). For instance, in applying FAP *stretch_r_cornerlip* (Stretch right lip comer) to a face mesh, the farther the lip vertices are from the comer lip, the less they are affected by the FAP. [ok,] is a matrix that has as many rows as the number of vertices of face model and as many columns as the number of FAP (68). To express how the application of FAP *i* updates the whole face model, the (1) can also be written as:

$$V(t) = V(0) + FAPU_i * A_i * x_i(t) \qquad (2)$$

where *V(t)* and *V(0)* represent the matrices of the coordinates of the vertices of the face model at time *t* and of the neutral face model respectively, A, represents the i-th column of the $[\alpha_{ki}]$ matrix. The coefficients α_{ki} have been estimated in the following way: called d_{ki} the distance between vertex v_k and the feature point associated to FAP *i*,

$$\alpha_{ki} = \exp(-d_{ki}) \quad \text{if } d_{ki} < T,$$
$$\alpha_{ki} = 0 \quad \text{if } d_{ki} \geq T_i$$

where *T* , is a threshold value suitable for FAP *i; T,* is chosen heuristically.

Afterwards, α_{ki} coefficients have been refined by means of subjective tests to achieve more natural and human-like facial deformations.

Other FAPs only involve the updating of a global parameter; for instance roto-translation of the head can be achieved modifying *the rotation* and *translation* values of the mesh representing the face model.

The algorithm for facial animation has been chosen to be very simple so that it can be executed in real time even on relatively low cost hardware platforms. The real time requirement is especially important when the face animation must be synchronised with an audio source and no delays are permitted.

4.1 Face Animation Driven by Speech

Face animation driven by synthetic speech will probably become more and more common in future because of the many interesting applications that can be developed. One of the main problems in this field, is the conversion of speech information (phonemes, prosody.. .) into animation parameters for the face models. In MPEG-4 [3], the class ttsFAPinterface defines the data structure for the interface between the speech synthesizer and the phoneme-to-FAP converter:

```
class ttsFAPinterface {
        phonemeSymbol;
        phonemeDuration;
        f0Average;
        stress;
        word-begin ;
}
```

where phonemeSymbol is expressed using the International Phonetic Alphabet (IPA)[2].

An MPEG-4 TTS system that takes a text string as input, must be able to generate the corresponding set of phonemes. These phonemes are mapped to the corresponding visemes (being a *viseme* the visual associated counterpart of a human oral sound) that, in turn, are finally converted into sets of FAPs. The visemes express lips /jaw position corresponding to each phoneme that is pronounced by the speaker. In normal conditions there is a strong relationship between aural and visual stimuli because there are precise dependence rules between articulator gesture, shape of the vocal duct and structure of the acoustic signal [5], [6], [7].

4.1.1 Phonemes to FAPs Conversion

The mapping between phonemes and visemes has been performed grouping together different phonemes in the same viseme, for instance: /p, b, m/, If, v/, It, d/, /s, z, dz, ts/, /tʃ, dg, ʃ, ŋ/, /n, 1, r, k/. The implemented set of visemes is suitable for Italian phonemes and is made of 11 visemes corresponding to consonants and 12 corresponding to vowels (a distinction between stressed and unstressed vowels and between open and closed /e/, /o/ has been performed).

Visemes can be characterised by a set of 4 macro-parameters [8], that are Jaw Opening, Lip Opening Height, Lip Opening Width and Lower Lip Protrusion. Each visemes is therefore expressed as a 4-ple that gives the intensities of the macro-parameters. Finally, each macro-parameter is converted into FAPs.

On the basis of the description of the most significant visemes it is therefore possible to drive the facial animation by composing the sequence of FAPs that make up the visemes. In Figure 4-1, three visemes (associated to phonemes /u/, /o/, /a/) are shown.

4.1.2 Synchronisation of Face Animation and Synthetic Speech

To achieve a pleasant result, the animation of a face model must be very well synchronised with the synthetic speech generated by the TTS module. Studies on this matter [9] have shown that, on average, if the time distance between lip motion and audio signal is in interval (-40 ms ÷ 120 ms), the user cannot perceive any misalignment. Lip syncbronisation is still acceptable if it is in (-90 ms ÷ 180 ms) interval.

The TTS Phonetic Module does not generate the phonemes continuously but in sets corresponding to the sentence (or part of the sentence) read from the input text. Hence, phonemes are not synthesised as soon as they are produced, but they are stored and "played" subsequently.

[2] http://www.arts.gla.ac.uk/IPA/ipa.html

The face animation system must therefore keep track of which phoneme is being played at a given time by means of a time signal received from the audio subsystem. The synchronisation mechanism is made in such a way that it can slow down the face animation if this is too fast or skip some visemes if the animation is late with respect to the speech.

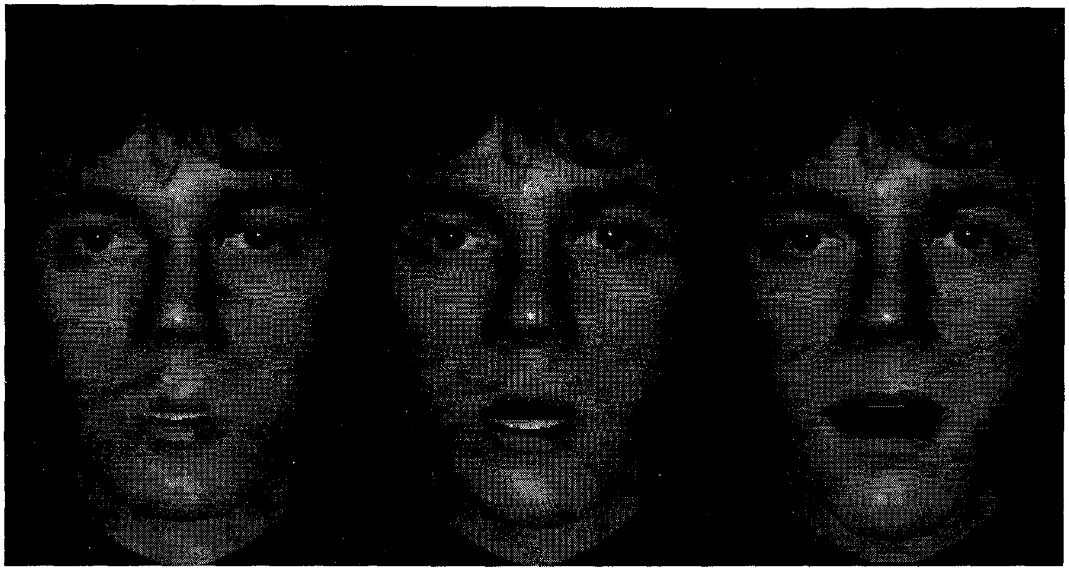

Figure 4-1 Visemes associated to phonemes /u/, /o/, /a/

4.2 Predefined vs. Downloaded Face Model

With regard to the facial models to be used during rendering and presentation, the following two alternatives are available:

- the model is predefined
- the model is transmitted as a polygon mesh and (optionally) associated texture

In the former case, a face model stored in the viewer can be used; this model can be calibrated and texture mapped as explained above (see Chapter 3).

In the latter case, a completely new face model can be transmitted through the incoming stream together with a set of Facial Definition Tables (FDTs) that tell the decoder how to apply FAPs on the model (see [1]). The transmission of a new model is typically precedes FAPs, so that the decoder can start animating a new face model, as soon as FAPs are received. FDTs define either global modification of the model (rotations, translations, scaling) or local deformations. In the latter case, FDTs express how to move the vertices of the model according to the received FAP; the motion of the vertices is approximated by a piecewise linear trajectory. It must be noted that the motion of the vertices can be as smooth as desired if the intervals in which the trajectory is linear are small enough.

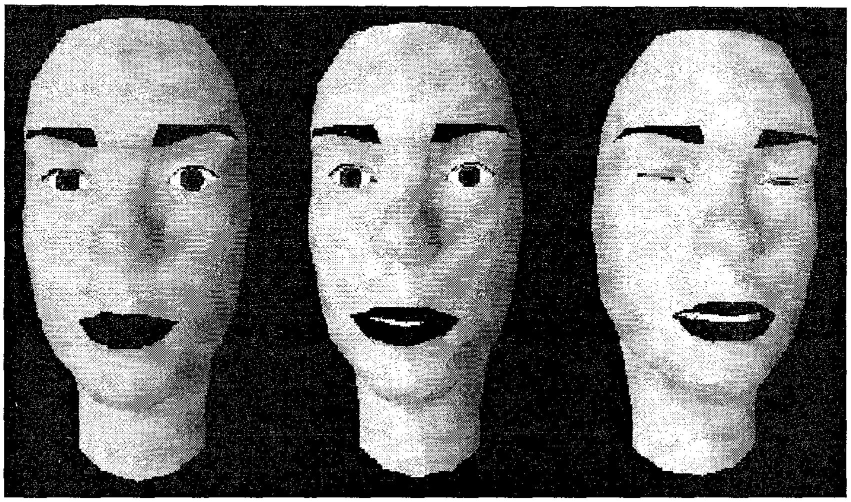

Figure 4-2. Animation of a downloaded model

4.3 Implementation of the Facial Animation System

The demo application named **JOE – Join Our Experience™-**, runs in a web browser equipped with a VRML plug-m. Figure 4-3 shows the application architecture (ellipses and rectangles represent interfaces and applications, respectively while full and dotted lines represent data and control flows, respectively).

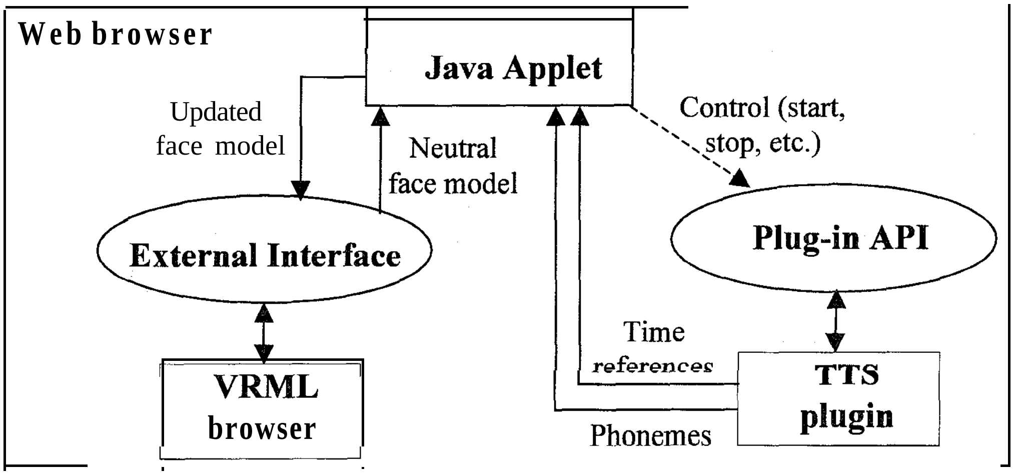

Figure 4-3. Application architecture

The application is based on "embedded" objects in an HTML page. The type of contents activates the following components:

- VRML plug-in for the rendering and the visualisation of the facial model;
- Java applet that communicates with the VRML viewer and applies local deformations and global transformations (translations and rotations) to the wire-frame model;
- TTS plug-in (built on CSELT's ***Eloquens®*** [3])

[3] http://voce.cselt.it/ufa/ufaUK/ufaProdotti/ufaSintesidatesto/

The Java program, through the External Authoring Interface[4], communicates with the VRML plug-in and can read and modify the 3D model, drive the timing of actions.

If some text is provided to the application, the TTS plug-in is instantiated and run to generate the speech and the corresponding phonemes. These phonemes are used by the Java applet to animate the face model; the time references generated by the TTS are used as master clock of the system to keep the synthetic speech and the face animation synchronised.

5. Conclusions

This work, done in SNHC FBA context, represents a prototype of a "talking head" in which a good degree of photo-realism and likelihood with respect to a human speaker has been achieved. For the implementation of the prototype, existing technologies have been used and the state of the art (VRML, Java, and web browser plug-ins) has been exploited. The developed system represents an almost complete implementation of MPEG-4 FBA specification.

For facial animation, simple algorithms have been preferred to more complex ones in order to be able to execute the application in real time and synchronised with an audio source on relatively low cost platforms as PCs.

For facial calibration the requirements are different: it is an off-line process, and it must ensure very precise model adaptation to the human face. Hence more complex and time-consuming algorithms can be used.

The implementation of the facial animation system can run in real time on medium/high level PCs (at least Pentium 166 MHz) with an OpenGL graphics board; hardware acceleration increases the speed of the rendering of the face models so that the correct frame rate can be ensured.

For test purposes, a Pentium 166 MHz 64 MB with a low cost OpenGL card (Leadtek WinFast 3D L2200 8MB) has been used. On this platform mainly two kind of tests have been performed: face animation driven by TTS phonemes and by FAPs read from a local file. In first case the application animates the face model at about 12 visemes/sec (this corresponds to an average duration of 80 ms per phoneme). In second case, of course the face animation can be slowed down if the face model is placed in a complex scene because of the rendering time of the 3D objects.

Tests executed on higher performance platforms have shown that it would be possible to use the developed application to animate more than one face model at the same time. This could allow the development of appealing applications like virtual meetings, virtual teleconference, sharing of virtual worlds where each user is represented by a face model and so on.

As an application of the developed system, a mail teller program has been written that reads aloud E-mail messages. The synthetic character used for reading has the countenances of the sender of the messages.

6. Future Work

Starting from the achieved results, the performance of the system is going to be strengthened and improved keeping in mind that the final objective is to line up the system to the MPEG-4 SNHC model.

In order to reach that goal, the following activities are planned:

Harmonise calibration techniques to MPEG-4

The target application will be able to calibrate the predefined facial model according to the parameters defined by MPEG-4, by adapting the face models to sets of 3D feature points or to arbitrary meshes.

Improve and personalise the synthetic characters

The synthetic character in the current version is not complete: the back of the head, the ears and the tongue are missing. In next version, face models are going to have all these parts so that all 68 FAPs can be applied. This will complete the MPEG-4 compliance as far as the face model animation is concerned.

In order to achieve a higher photo-realism and resemblance of a virtual character to a given human being, improvements will be made to personalise the voice, the countenances and the animation of the face models.

Improve animation smoothness

In current version, the described application animates the face animation applying one viseme a time. This leads to rather jerky movements because of the sudden transition between visemes.

A much better approach would

- consider two successive visemes
- generate intermediate positions between them (e.g. by means of linear interpolation)
- use these inter-visemes positions in face animation to perform a smoother transition between visemes.

Handle coarticulation

This activity can be seen as a refinement of previous one in which viseme interpolation is based on the observation of real human animation and not on mathematical rules.

ufaEloquens2000/ufaEloquens2000.html

[4] http://vrml.sgi.com/moving-worlds/spec/ExternalInterface.html

7. Acknowledgements

The authors would like to thank Marco Di Pasquale for implementing the visemes-to-FAPs mapping, for the work done to improve the speech / animation synchronisation and for all the other improvements in the system. We are grateful to Stefano Lazzaretto, for helping in the integration of the TTS module in the face animation application, for his invaluable suggestions, for the many hours spent to implement the requested routines. Finally, we would like to thank AT&T for providing a face model that we used for the tests on downloaded models.

8. References

[1] MPEG Systems Group “Text for FCD 14496-1 Systems”, ISO/IEC JTC1/SC29/WG11 N2201, April 1998.

[2] MPEG Video Group “Text for FCD 14496-2 Video”, ISO/IEC JTC1/SC29/WG11 N2202, April 1998.

[3] MPEG Audio Group “Text for FCD 14496-3 Audio”, ISO/IEC JTC1/SC29/WG11 N2203, April 1998.

[4] J. D. Foley, A. van Dam, S. K. Feiner, J. F. Hughes, “Computer Graphics Principles and Practice, Second Edition in C”, Addison Wesley, 1995.

[5] A. Q. Summerfield, “Audio-Visual Speech Perception, Lipreading and Artificial Stimulation”, in M.E. Lutman, M.P.Haggard, ***Hearing Science and Hearing Disorders,*** Academic Press, London, U.K., 1983, pp. 131-182.

[6] A. Q. Summerfteld, “Use of Visual Information for Phonetic Perception”, ***Phonetica,*** 1979, Vol. 36, pp. 3 14-33 1.

[7] A. Q. Summerfield, “Some Preliminaries to a Comprehensive Account of Audio-Visual Speech Perception”, in B. Dodd, R.Campbell, ***Hearing by Eye: the*** Psychology of Lip-reading, Lawrence Erlbaum Ass. Publ., Hillsdale, New Jersey, 1987, pp. 3-52.

[8] E. Magno Caldognetto, P. Cosi, “Lips and Jaw Movements for Vowels and Consonants: Spatio-Temporal Characteristics and Bimodal Recognition Applications”, CSRF Centro di Studio per le Ricerche di Fonetica - CNR, Padova, Italy.

[9] S. Rihs, “The Influence of Audio on Perceived Picture Quality & Subjective Audio-Video Tolerance.“, *Proc.* ***MOSAIC Workshop on Advanced Methods for the Evaluation of Television Picture Quality,*** Eindhoven, Netherlands, September 1995, pp. 133-137.