The Facial Animation Engine: towards a high-level interface for the design of MPEG-4 compliant animated faces

Fabio Lavagetto[1], Roberto Pockaj[2]

DIST – University of Genova, Italy
[1] fabio@dist.unige.it, [2] pok@dist.unige.it

Abstract

In this paper we propose a method for implementing a high-level interface for the synthesis and animation of animated virtual faces that is in full compliance with MPEG-4 specifications. This method allows us to implement the Simple Facial Object profile and part of the Calibration Facial Object Profile.
In fact, starting from a facial wire-frame and from a set of configuration files, the developed system is capable of automatically generating the animation rules suited for model animation driven by a stream of FAP (Facial Animation Parameters). If the calibration parameters (feature points and texture) are available, the system is capable to exploit this information for modifying suitably the geometry of wire-frame and for performing its animation by means of calibrated rules computed "ex-novo" on the adapted somatics of the model.
Evidence of the achievable performance is reported at the end of the paper by means of pictures showing the capability of the system to reshape its geometry according to the decoded MPEG-4 facial calibration parameters and its effectiveness in performing facial expressions.

1. Introduction

MPEG-4 activities were started in 1993 following the previous successful experiences of MPEG-1 (for applications of storage and retrieval of moving pictures and audio on/from storage media) and MPEG-2 (for broadcasting application in digital TV). MPEG-4 has as main mandate the definition of an efficient system for encoding Audio-Visual Objects (AVO), either natural or synthetic [8]. This means that, towards the revolutionary objective of going beyond the conventional concept of the audio/visual scene as composed rigidly of a sequence of rectangular video frames with associated audio, MPEG-4 is suitable structured to manage any kind of AVO, with the capability to compose them variously in 2D and 3D scenes. The heart of MPEG-4 is, therefore, composed of the architecture of systems, responsible for the definition of the respective position of objects composing the scene, of their behaviour and of their interactions. This information is described through a hierarchical description within the so-called "scene graph". A typical, frequently used, scheme of a MPEG-4 scene is represented in Figure 1.
According to the scheme sketched in Figure 2, the various audio-visual objects composing the scene can be retrieved or originated either locally or remotely. An "elementary stream", associated with each object, is then multiplexed and transmitted. The receiver applies the inverse process providing to the compositor all the necessary information on objects and on the scene graph, which enables the reconstruction of the audio-visual sequence.
Also in this framework, synthetic objects like human faces can be introduced. In fact, among the multitude of possible objects that can be created artificially at the computer, MPEG has chosen to dedicate specific room to this class of objects that are considered to play a role of particular interest in audio-visual scenes. Therefore, the necessary suitable syntax has been standardized in MPEG-4 for the definition and animation of synthetic faces.
This paper describes a partial implementation of the MPEG-4 specifications for the adaptation and animation of 3D wire-frames suited to model human faces with respect to the reproduction of both their static characteristics (realism in adapting the model geometry to the somatics of any specific face) and dynamic behaviour (smoothness in rendering facial movements and realism in performing facial expressions).
Let us introduce the concept of "tool" that is defined as the ensemble of algorithms representing partially or totally the processing necessary to implement a given application. A quantity of such tools has been defined in MPEG-4 so far in order to provide all the computational blocks necessary to implement the largest set of applications. This is true also in the case of face animation for which three kinds of decoding tools have been defined for the configuration and animation

of a synthetic human character, capable of decoding the standard facial parameters described in the following paragraphs.

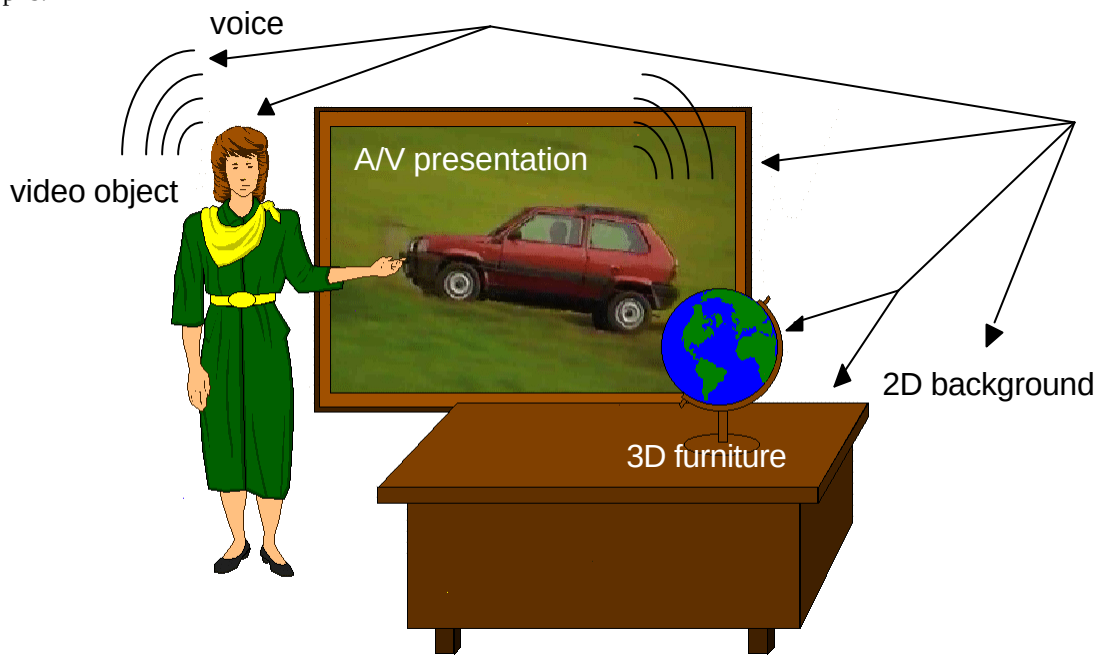


Figure 1: Typical MPEG-4 scene with its associated "scene-graph"

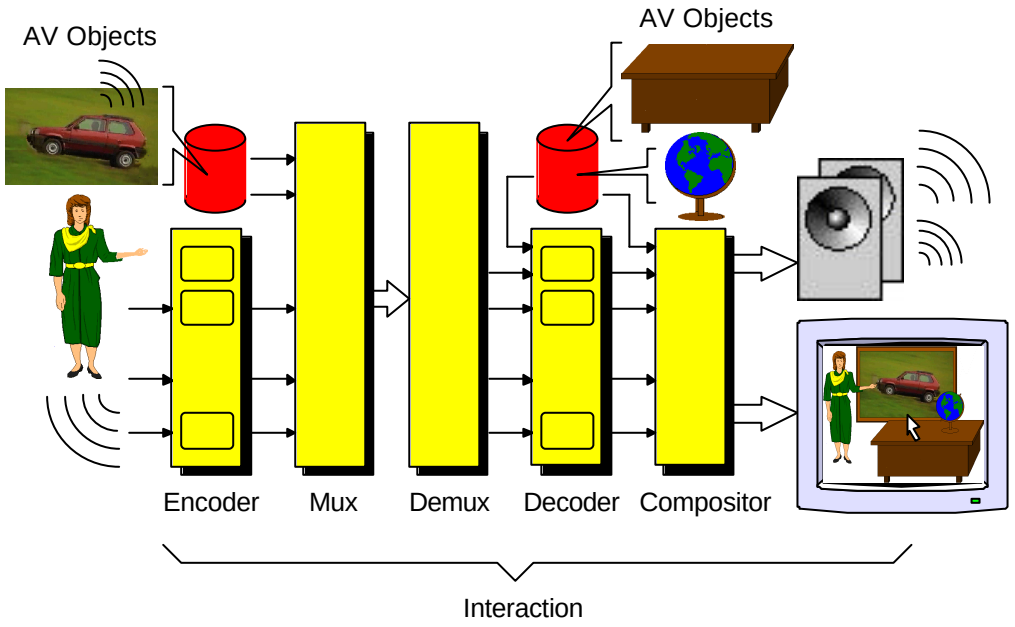


Figure 2: Scheme of the MPEG-4 System

FAP (Facial Animation Parameters) are responsible for describing the movements of the face, either at low level (i.e. displacement of a specific single point of the face) or at high level (i.e. reproduction of a facial expression). In few words, FAP represent the proper animation parameter stream.

FDP (Facial Definition Parameters) are responsible for defining the appearing of the face. These parameters can be used either to modify (though with various levels of fidelity) the shape and appearing of a face model already available at the

decoder, or to encode the information necessary to transmit a complete model together with the criteria that must be applied to animate it. In both cases the animation of the model is described only by the FAP. FDP, to the contrary, are typically employed only when a new session is started.

FIT (FAP Interpolation Table) is used to reduce the FAP bitrate. By exploiting the a priori knowledge of the geometry and dynamics of human faces, it is possible to deduce the value of some FAP based on the knowledge of some others. Even if these inference rules are proprietary to the model used by the decoder, the application of some specific rules can be forced by encoding them in the bitstream through FIT.

After the concept of "tool", it is necessary to introduce the definition of "object profile" whose detailed description is very long and is beyond the scope of this paper. An object profile describes the syntax and the decoding tools for a given object. An MPEG-4 decoder compliant with a given object profile must necessarily be able to decode and use all the syntactic and semantic information included in that profile. This mechanism allows one to classify the MPEG-4 terminals by grouping object profiles into composition profiles that define the terminals performance. Even though the discussion is still open, MPEG-4 will define very likely three different Facial Animation object profiles [3]:

1) **Simple Facial Animation Object Profile**: It is mandatory for the decoder to use FAP with the option of considering or ignoring all the other facial information encoded in the bitstream. In this way, the encoder has no knowledge of the model that will be animated by the decoder and cannot in any way evaluate the quality of the animation that will be produced.

2) **Calibration Facial Animation Object Profile**: In addition to the modality defined by the Simple profile, the decoder is now forced also to use a subset of the FDP. As introduced before, FDP can operate either on the proprietary model of the decoder or, alternatively, on a specific model (downloadable model) encoded in the bitstream by the encoder. In this profile the decoder is compelled to use only the FDP subset, namely "feature points", texture and texture coordinates, which drives the reshaping of the proprietary face model without requiring its substitution. The feature points, described in detail later on in the paper, represent a set of key-points on the face that are used by the decoder to adapt its generic model to a specif geometry and, optionally, to a specific texture. The decoder must also support FIT.

3) **Predictable Facial Animation Object Profile**: In addition to the Calibration profile, all of the FDP must be used, included those responsible of downloading the model from the bitstream. By using a specific combination of FDP (which includes the use of Facial Animation Table), the encoder is capable of completely predicting the animation produced by the decoder.

For a detailed description of all the parameters so far introduced, please refer to the Visual Committee Draft [5] and Systems Committee Draft [4].

The synthetic facial model presented in this paper is compliant with MPEG-4 specifications and is characterized by full calibration and animation capabilities. Many different application scenarios have been proposed that are ready to integrate this novel technology into consolidated multimedia products and services. As an example, challenging exploitations are expected in movie production for integrating natural and synthetic data representation, in interactive computer gaming, in advanced "human" interfaces and in a variety of multimedia products for knowledge representation and entertainment. At the 44th MPEG-4 meeting, held in Dublin in July 1998, the facial model described in this paper has been given to the MPEG-ISG (Implementation Studies Group) for testing candidate algorithms for evaluating the complexity introduced by facial rendering. Sample images and demo movies of the facial model will be available at the web site http://www-dsp.com.dist.unige.it.

A focused summary of the activity carried out by the MPEG-4 "ad hoc" group on Face and Body Animation (FBA) is given in section 2 together with a description of the standardized facial parameters. A detailed description of the facial model implemented by the authors is provided in section 3, while a few preliminary examples of technology transfer and application are described in section 4. Final conclusions are drawn in section 5.

2. Facial Animation Semantics

2.1 The Neutral Face

The basic concept governing the animation of the face model is that of “Neutral Face”. In fact, all the parameters that drive the animation of the synthetic face indicate relative displacements and rotations of the face with respect to the neutral position.

Let us define the concept of neutral face, corresponding to the face posture represented in Figure 3:

?? the coordinate system is right-handed;
?? head axes are parallel to the world axes;
gaze is in direction of the Z axis;
?? all face muscle are relaxed;
?? eyelids are tangent to the iris;
?? pupil diameter is 1/3 of iris diameter;
?? lips are in contact; the line of the lips is horizontal and at the same height of lip corners;
?? the mouth is closed and the upper teeth touch the lower ones;
?? the tongue is flat, horizontal with the tip of the tongue touching the boundary between upper and lower teeth.

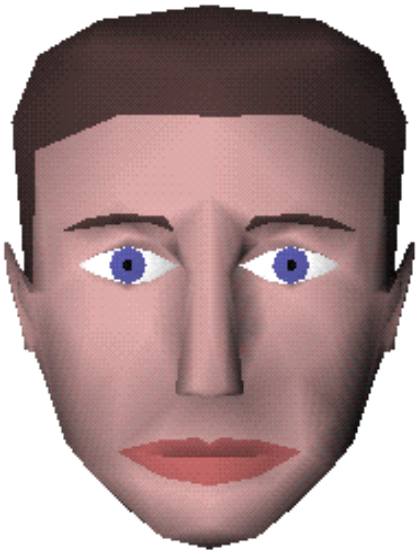


Figure 3: The Neutral Face

2.2 The feature points

The “feature points”, which define relevant somatic points on the face, represent the second key semantic concept. Feature points are subdivided in groups, mainly depending on the particular region of the face to which they belong. Each of them is labeled with a number identifying the particular group to which it belongs, and with a progressive index identifying them within the group. The position of the various features of the face is shown in Figure 4 while in Table 1 the characteristics, location and some recommended constraints are listed for the feature points in the eyebrow region.

Feature points		**Recommended location constraints**		
#	**Text description**	**x**	**y**	**z**
...	...	...	...	...
4.1	Right corner of left eyebrow			
4.2	Left corner of right eyebrow			
4.3	Uppermost point of the left eyebrow	(4.1.x+4.5.x)/2 or x coord of the uppermost point of the eyebrow		
4.4	Uppermost point of the right eyebrow	(4.2.x+4.6.x)/2 or x coord of the uppermost point of the eyebrow		

4.5	Left corner of left eyebrow			
4.6	Right corner of right eyebrow			
...	...	...	...	...

Table 1: Feature Points Description Table

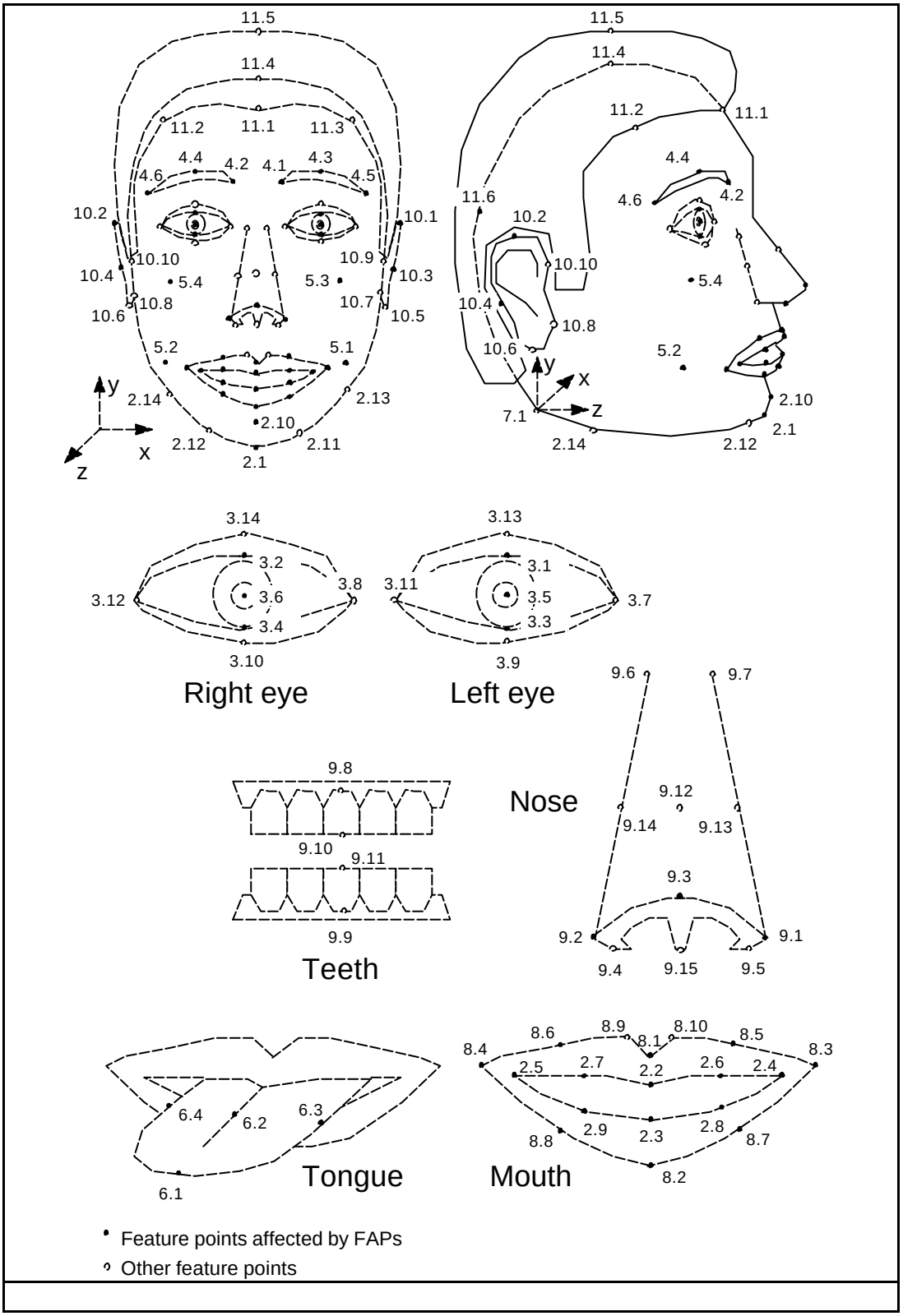


Figure 4: Feature Points

2.3 The Facial Animation Parameters (FAP) and their measurement units (FAPU)

While all the feature points contribute to define the appearing of the face in the Calibration Facial Animation Object Profile, some of them are also associated with the animation parameters. These latter, in fact, define the displacements of the feature points with respect to their positions in the neutral face. In particular, a FAP encodes the magnitude of the feature point displacement with respect to one of the three Cartesian axes, except that some parameters encode the rotation of the whole head or of the eyeball.
The animation parameters are described in a table whose section related to the eyebrows region is reported below in Table 2. In each table row, the number and the name of the FAP are reported together with a textual description of the FAP, the corresponding measure unit, a notation to identify if they mono-directional or bi-directional, an indication of the positive direction of the motion, the feature point affected by the FAP (by identifying the group and the sub-group it belongs to) and the quantization step.

#	FAP name	FAP description	units	Uni / Bidir	Positive Motion	Grp	FDP subgrp num	Quant step size
...	...	...	...	...	...	...	...	...
31	raise_l_i_eyebrow	Vertical displacement of left inner eyebrow	ENS	B	up	4	1	2
32	raise_r_i_eyebrow	Vertical displacement of right inner eyebrow	ENS	B	up	4	2	2
33	raise_l_m_eyebrow	Vertical displacement of left middle eyebrow	ENS	B	up	4	3	2
34	raise_r_m_eyebrow	Vertical displacement of right middle eyebrow	ENS	B	up	4	4	2
35	raise_l_o_eyebrow	Vertical displacement of left outer eyebrow	ENS	B	up	4	5	2
36	raise_r_o_eyebrow	Vertical displacement of right outer eyebrow	ENS	B	up	4	6	2
37	squeeze_l_eyebrow	Horizontal displacement of left eyebrow	ES	B	right	4	1	1
38	squeeze_r_eyebrow	Horizontal displacement of right eyebrow	ES	B	left	4	2	1
...	...	...	...	...	...	...	...	...

Table 2: Facial Animation Parameters Description Table

Description		FAPU value
IRISD0 = 3.1.y - 3.3.y = 3.2.y - 3.4.y	IRIS Diameter (by definition it is equal to the distance between upper and lower eyelid) in neutral face	IRISD = IRISD0 / 1024
ES0 = 3.5.x - 3.6.x	Eye Separation	ES = ES0 / 1024
ENS0 = 3.5.y - 9.15.y	Eye - Nose Separation	ENS = ENS0 / 1024
MNS0 = 9.15.y - 2.2.y	Mouth - Nose Separation	MNS = MNS0 / 1024
MW0 = 8.3.x - 8.4.x	Mouth - Width Separation	MW = MW0 / 1024
AU	Angular Unit	10^{-5} rad

Table 3: Facial Animation Parameters Units Description Table. Symbols and formulas in the lefthand column are referred to the facial feature points shown in Figure 4. Extensions .x and .y stand for the horizontal and vertical coordinate of the feature point, respectively.

The magnitude of the displacement is expressed by means of specific measure units, called FAPU (Facial Animation Parameter Unit). Each FAPU represents a fraction of a key-distance on the face (as an example 1/1024 of the mouth width), except for FAPU used to measure rotations; this allows us to express FAP in a normalized range of values that can be extracted or reproduced by any model. A description of FAPU is reported in Table 3 and in Figure 5.

The method for interpreting the value of FAP is therefore quite simple. After decoding a FAP, its value is converted in the measure system of the model that must be animated, by means of the appropriate FAPU. Then, the corresponding feature point is moved, with respect to its neutral position, of the computed displacement. Since a FAP defines only how to compute on of the three coordinates of a specific point on the face, each FAP decoder must be able to determine autonomously the other two coordinates of the feature point which are not specified by the FAP and the coordinates of all those vertices which compose the wire-frame of the face which, being in the proximity of the feature point, are influenced by its motion. It is apparent how the intelligence of the decoder determines the implementation of reliable and realistic mechanisms for facial animation. In the next section, a technique is proposed and described for implementing a "Facial Animation Engine", capable of animating a generic facial wire-frame by exploiting uniquely the knowledge of its topology and of the semantics of its vertices.

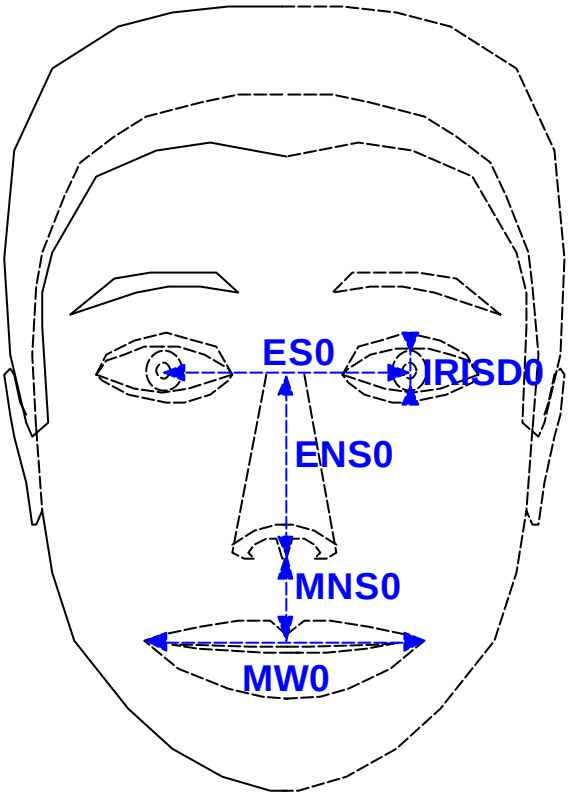


Figure 5: Facial Animation Parameters Units

3. The Facial Animation Engine

The core mechanisms responsible of the facial animation are based on software procedures that compose a high level interface for the implementation of 3D models of animated faces. In particular, this interface allows the implementation of a decoder compliant with the Simple Facial Animation Object Profile and with the Calibration Facial Animation Object Profile.The Facial Animation Engine (FAE) is capable of animating a facial wire-frame starting from its geometric description and from the associated semantic information.

The FAE is independent of the shape and size of the face model that must be animated; it is capable of defining the animation rules for each FAP whatever is the wire-frame on which it operates. For this reason, it is completely compliant with the specifications of the Calibration Profile since its performance is guaranteed when acting on either the proprietary model left as it is or when acting on the proprietary model reshaped according to the calibration FDP. A high-level description of the FAE is shown in Figure 6.

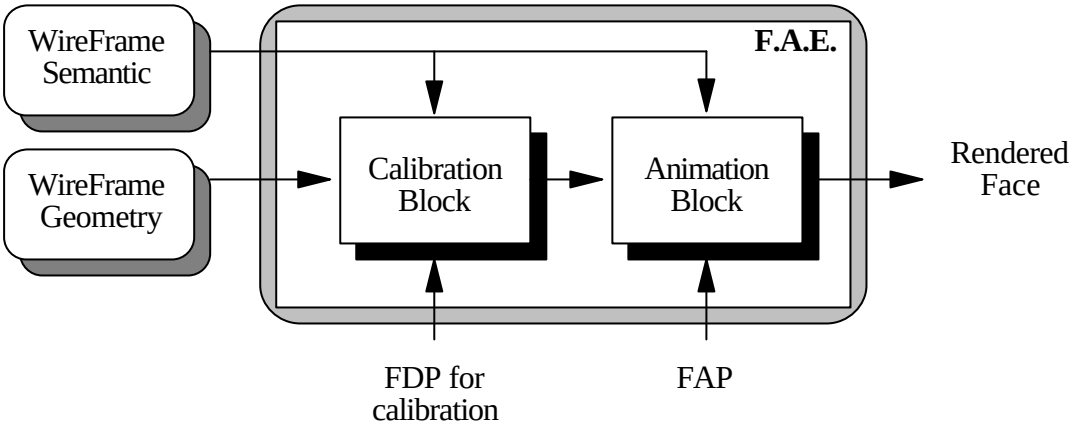


Figure 6: Block diagram of the Face Animation Engine

3.1 The Animation Block

Let us start with the description of the animation procedures assuming, as said before, its independence from the model geometry. We will then proceed with the description of the calibration module, which is activated only in case calibration parameters are encoded in the bitstream.
The wireframe geometry consists of a conventional representation VRML-like of a net of polygons. It contains the coordinates of the vertices and the topology of the wire-frame representing the face, together with the color information associated to each polygon.
The wireframe semantics consists of high-level information associated to the wireframe like:

?? list of wire-frame vertices which have been chosen as “feature points” (information eventually used by the calibration module);
?? list of vertices affected by each specific FAP;
?? definition of the displacement region for each FAP (i.e. region of the face affected by the deformation caused by a specific FAP).

All this information is expressed in a simple manner, as a function of the wire-frame vertices, so that these semantic definitions appear to be immediately comprehensible starting with any generic wire-frame.
Besides the list of the points which are moved by each specific FAP, the Face Animation Engine operates on a table that associates with each FAP a pre-defined movement that is applied to the associated vertices. It is apparent that different FAP can use the same pre-defined movement so that it is reasonable to assume significantly fewer movements than FAP. Some examples of pre-defined movements are reported in the following:

?? translation parallel to X, Y or Z (only one coordinate of the vertex is modified by the displacement);
?? translation on planes parallel to [X,Z] or to [Y,Z];
?? proper rotation around an axis parallel to X, Y or Z

Based on the information described above, vertices associated with FAP and pre-defined movements, the animation engine automatically computes the animation rules.
To describe this procedure, let us introduce an example. Let us consider, among the many movements that can affect the left eyebrow, that one which is controlled by FAP31 (raise_l_i_eyebrow), which controls the vertical displacement of the eyebrow internal region.
From the semantics associated to the wire-frame, the FAE recognizes the vertices that are affected by the displacement controlled by FAP31 and identifies the region onto which the displacement is mapped. In addition, the FAE also knows that the pre-defined movement associated to FAP31 is the vertical translation along Y + Z.

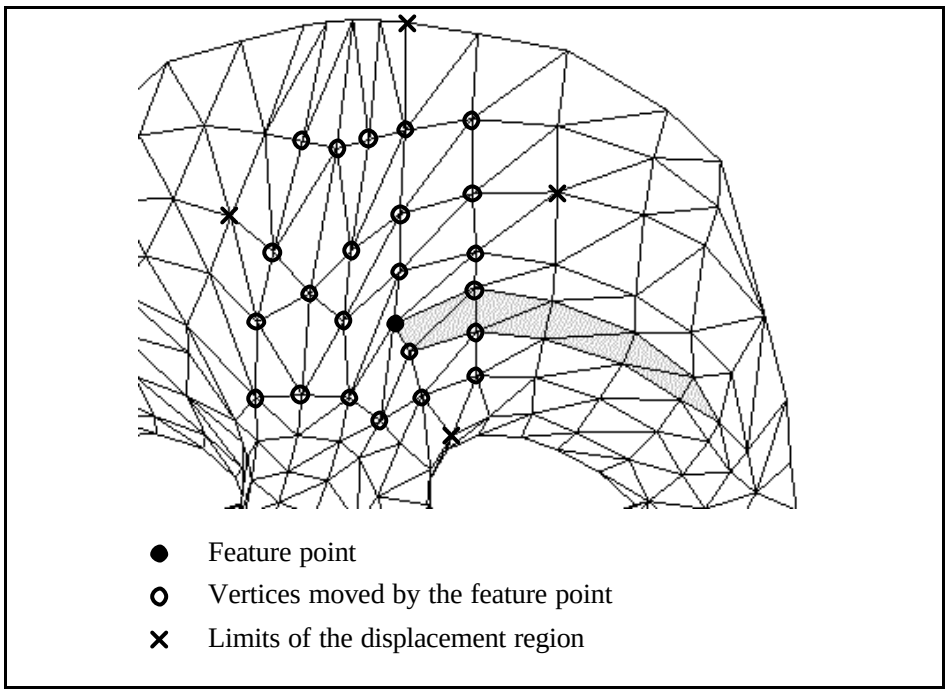


Figure 7: Information necessary to compute the animation rule controlled by FAP31.

In Figure 7 the feature point associated to FAP31 is identified on the wire-frame region around the left eyebrow, together with the vertices which are affected by the pre-defined movement associated with FAP31 and the bounding points (vertices) which limit the domain for the effects produced by the movement itself.
For each vertex which must be moved, the FAE computes a specific weight that defines the magnitude of its displacement with respect to the feature point. Weights are defined as a function of their distance from the border of the displacement domain and a detailed description of their computation is reported in a following section of the paper devoted specifically to the pre-defined movements.
Once that a weight is associated with each vertex, the specific pre-defined movement is applied accordingly. In the case of FAP31, as said before, a vertical translation along Y+Z. However, along with its vertical movement, the eyebrow slides smoothly upwards on the skull, so that the z coordinate of the feature point and of the other affected vertices of the domain must vary in an anatomically consistent way, regardless of the particular shape of the skull.

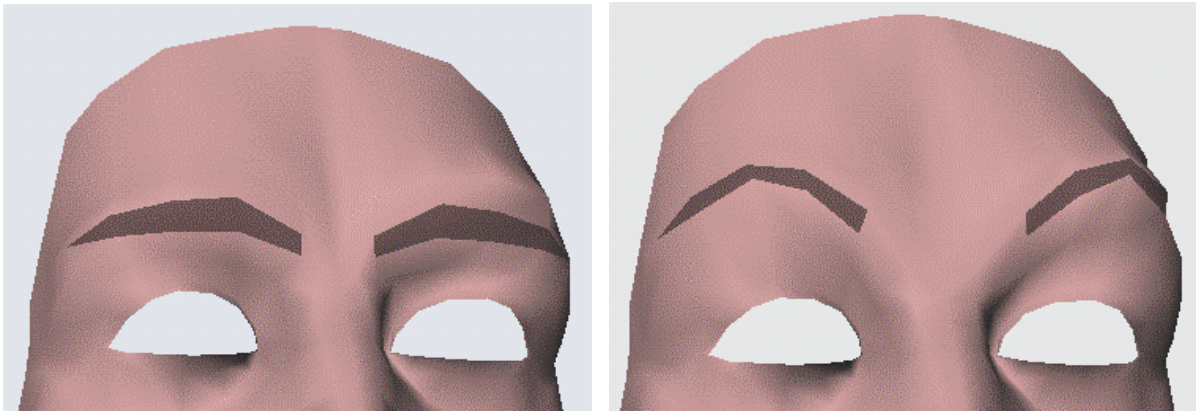


Figure 8: (Left) eyebrow squeezing and (right) eyebrow raising; automatically generated animation rules with the "Y+Z translation" pre-defined animation

Figures 8 and 9 show how, adopting the above described criteria, the movement of the eyebrow is reproduced anatomically (like in eyebrow squeezing) sliding over the skull surface without originating annoying artifacts. The animation rules generated automatically by the FAE take into account the local geometry of the face surface (as it is done in eyebrow raising) making it possible, for instance, to reproduce easily non-symmetrical movements.

It must be noticed how the displacement domain of each feature points can overlap one another (i.e. a vertex of the wire-frame not coinciding with a feature point can be affected by the movements associated with different feature points). Figure 9 shows the three displacement domains associated with the three FAP operating on the left eyebrow.

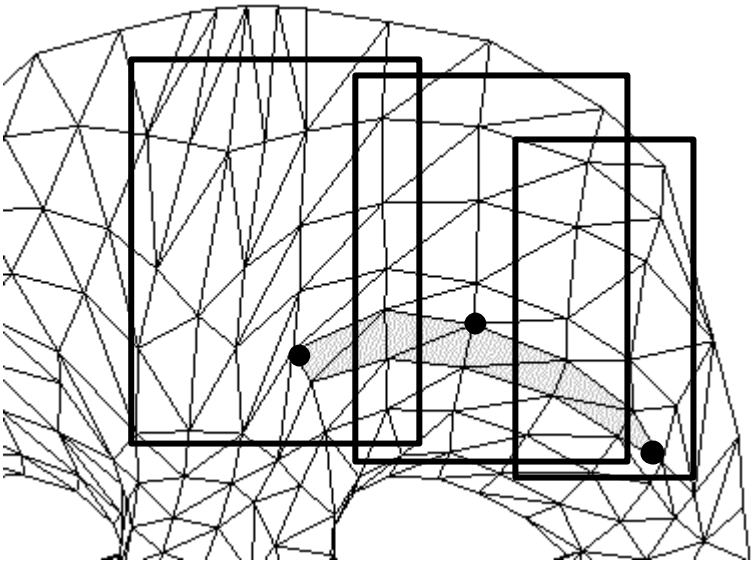


Figure 9: Displacement domains for FAP31, FAP33 and FAP35.

3.1.1 Description of the pre-defined movements:

?? Translation parallel to X, Y or Z

In case of translations parallel to one coordinated axis X, Y or Z, the displacement domain is defined by means of the semantic information associated with the wire-frame. For the computation of the weights, in particular, it is necessary to know the vertex identified as feature point, the vertices affected by the pre-defined movement and the vertices which define the displacement domain.

If we consider, as an example, the translation along Y, the weight associated with each vertex is computed as:

$$W_i \; ? \; \frac{? x_i}{? X} \; ? \frac{? y_i}{? Y}$$

where, with reference to Figure 10, P_i represents the i-th vertex for which the weight W_i is computed, F defines the corresponding feature point and X_{min}, X_{max}, Y_{min}, Y_{max} identify the bounding points which limit the displacement domain.

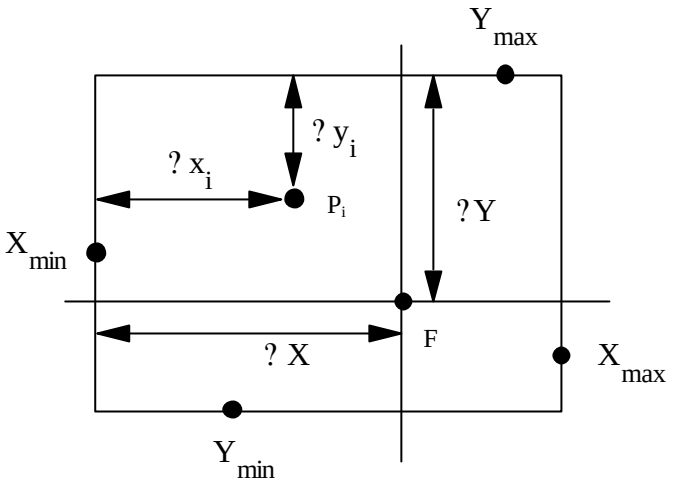


Figure 10: Algorithm for computing the weights associated to vertices.

The displacement of each vertex along the Y-axis will be computed as:

$$Displacement\_of\_feature\_point * W_i$$

?? Translation on planes parallel to [X, Z] or to [Y, Z]

With this type of movement, the computation of the weights is done in a way similar to what described for the translations along one single axis except for the fact that in this case also the z coordinate is affected by the movement.
By sampling the neutral face, an approximation of the 3D geometry of the head is obtained which is then used to constrain the trajectories of the feature points and of the related vertices to lie on its surface. Exploiting this 3D a priori knowledge, it is possible to apply vertex translations on planes parallel to [X, Z] or to [Y, Z] by supplying explicitly only the x or y coordinate, respectively, while retrieving an estimate of the z coordinate indirectly from the face surface approximation.

?? Proper rotation around an axis parallel to X, Y or Z

From the semantic information about the wire-frame, it is possible to identify the set of points affected by the rotation, the direction of the rotation axis and the specific vertex lying on it. This kind of movements reproduces rigid rotations of the entire wire-frame or only of parts of it.
This movement is applied both for those FAP which encode a direct rotation, like the rotation of the head, and for those that indirectly imply a rotation, for example the FAP3 encoding the aperture of the jaw. In the later case, the angle of jaw rotation is estimated starting from the geometry of the model and from the linear displacement of the associated feature point.

?? Weighted X rotation

This kind of movement determines the rotation of a set of vertices around the coordinated axis X of an angle which varies vertex by vertex and is expressed as a function of their linear displacement along the axis Y. Also in this case, the weights which define the Y displacement of each vertex are computed as in the case of the translation movements. This particular movement is applied to vertices affected by the displacement of more than one feature point (i.e. lying in the intersection of multiple displacement domains).

3.1.2 Examples of animation

In the following pictures we report some results obtained using the data available in the Test Data Set (TDS) of the Face and Body Animation (FBA) “ad hoc” group of MPEG-4. The Facial Animation Engine has been employed to animate two different models. “Mike”, which is proprietary design of DIST. It is very simple but complete since it includes all the standardized feature points including tongue, palate, teeth, eyes, hair and back of the head [9]. “Oscar” is derived from

the geoface model implemented by Keith Waters [11]; it is more complex but still incomplete in some parts. As shown in Figure 11, “Mike” is composed of 683 polygons including external and internal face structures, while “Oscar” employs 878 polygons for modeling only the frontal face surface. It is therefore evident how the animation and calibration quality must be evaluated also in dependence to the complexity of the model on which algorithms are applied. As the model complexity increases, in fact, higher geometrical resolution and improved rendering are achieved at the expense of heavier computation.
In Figure 11, the wire-frame of the two models is reproduced to stress the different levels of complexity.
All these animation results have been generated without exploiting any calibration information: “Mike” and “Oscar” are hypothetical proprietary models for a Face Animation Decoder.

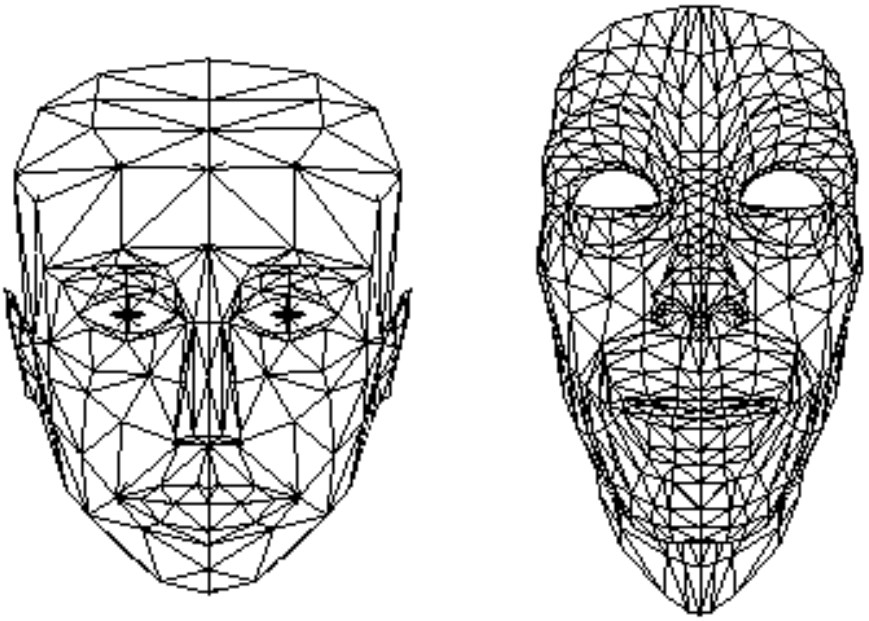
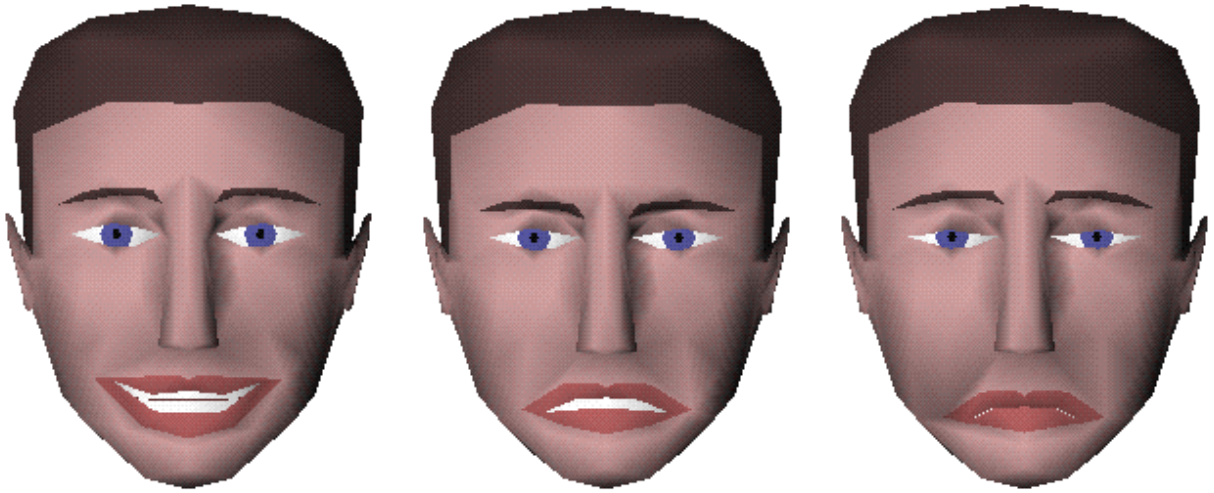


Figure 11: Wire-frame of “Mike” (left) and “Oscar” (right).



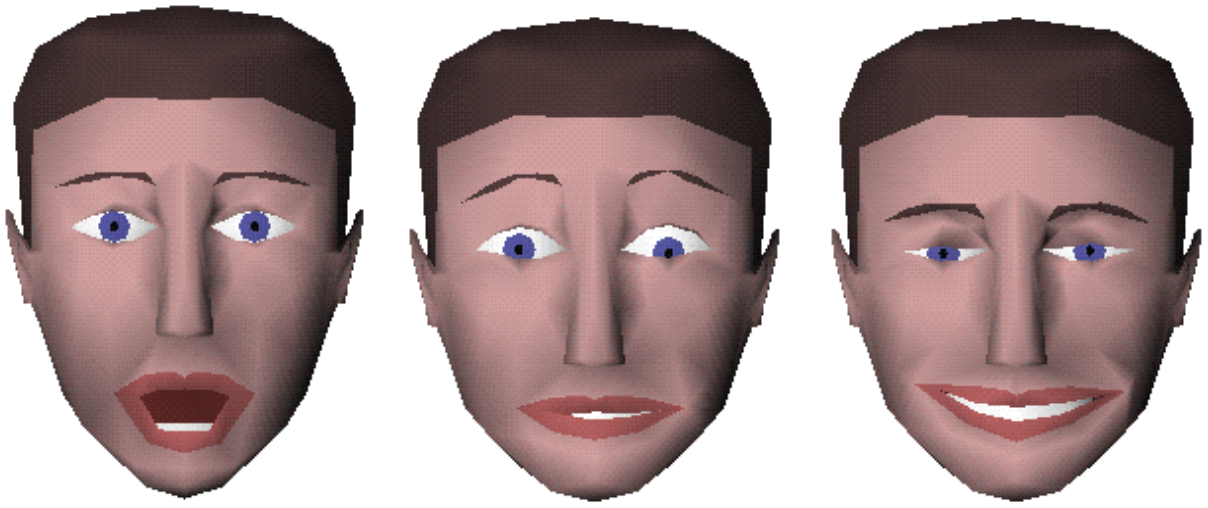
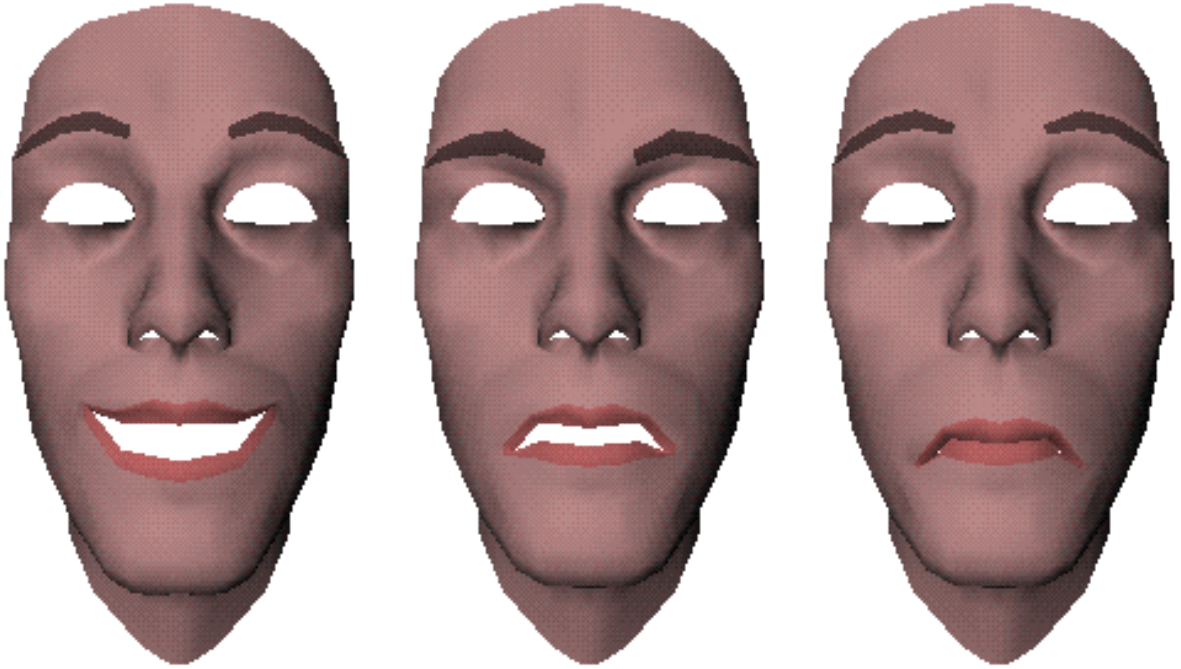


Figure 12: Facial expression of joy, anger, sadness and surprise extracted from the file “expressions.fap”, and eyebrow raising and smile extracted from the file “ Marco20.fap”, reproduced on the model “Mike”.



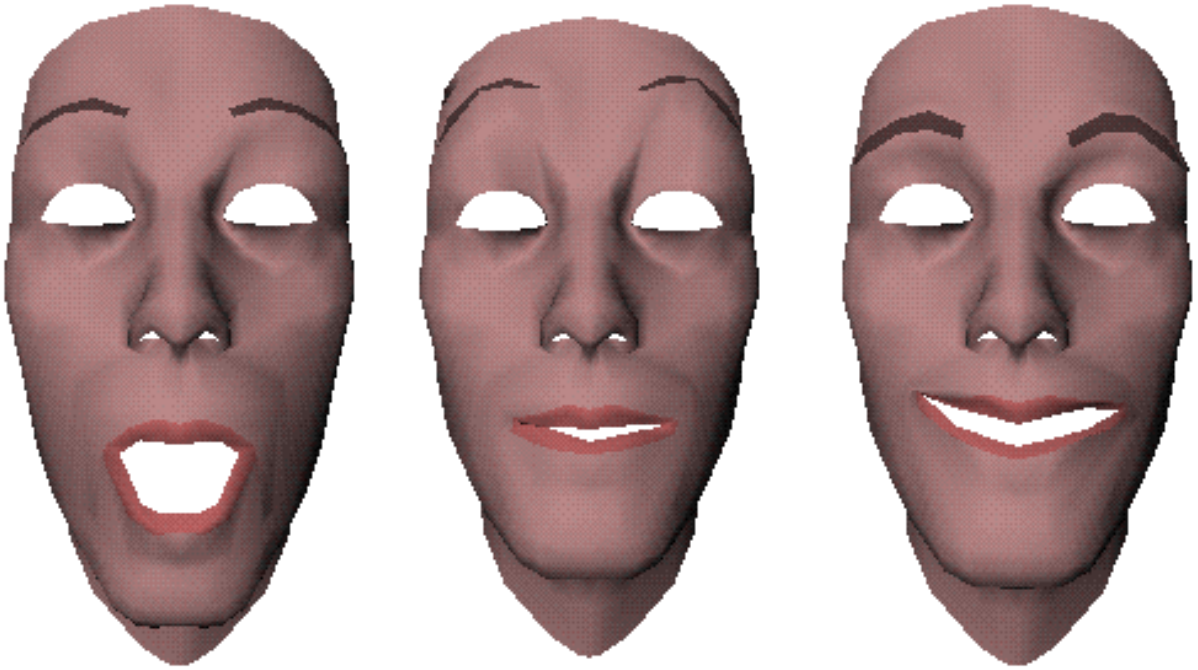


Figure 13: Facial expression of joy, anger, sadness and surprise extracted from the file “expressions.fap”, and eyebrow raising and smile extracted from the file “ Marco20.fap”, reproduced on the model “Oscar”.

3.2 The Calibration Block

In this paragraph the problem of model calibration is faced, introducing the use of FDP to reshape the geometry of the proprietary model and modify its appearance.

In the Facial Animation Object Profile specific parameters are standardized which enables the generic model available at the decoder to reshape its geometry to resemble a particular face. The information encoded through the FDP consists of a list of feature points and, optionally, a texture image with texture coordinates associated with the feature points.

The Calibration Facial Animation Object Profile requires the feature points of the proprietary model to be adjusted to correspond to the feature points encoded in the FDP, while no constraints are forced on the remaining vertices. The quality of the generated faces depends on how the unconstrained vertices are moved. We want to avoid distortion.The employed algorithm used to interpret the calibration FDP is based on the theory of Radial Basis Functions (RBF).

3.2.1 Multilevel calibration with RBF

Model calibration is performed by reshaping its geometry depending on the decoded calibration FDP in order to preserve the smoothness and somatic characteristics of the model surface. There are relatively few (typically 80) calibrated points relative to the global number of vertices in the wire-frame which, depending on its comp lexity, can have 1000 or more points. Model calibration can be considered as a problem of "scattered data interpolation" [15].

Let us assume the proprietary model to be composed of a finite set of points $X ? \{x_1,...,x_N\} ? ? ^3$, including a subset $S = \{s_1,\ldots,s_n\} ? X$, representing the ensemble of feature points. Let us associate with S a corresponding set $Y=\{y_1,\ldots,y_n\} ? ? ^3$ indicating the subset of calibration feature points transmitted by the encoder.
The problem to solve is to define an interpolation function $F: ? ^3 ? ? ^3$ such that:

$$F(x_i)=y_i, \quad 1 ? i ? n. \tag{1}$$

and that produces realistic results.

The approach that has been followed constructs the interpolation function model *F* as:

$$F(x) : ? \sum_{j?1}^{n} w_j ? (x ? x_j) \tag{2}$$

where w_j ? ? ^{3} represents the weight associated to the *j-th* function and the function ? : ? ^{3} ? ? is positive definite on ? ^{3} in the sense that for any finite set of points $S=\{s_1,\ldots,s_n\}$? ? ^{3} the matrix

$$A ? (? (x_k ? x_j))_{1? j,k? n} \tag{3}$$

is positive definite. Therefore the system is guaranteed to have a solution.

$$F(x_k) ? \sum_{j?1}^{n} w_j ? (x_k ? x_j) ? y_k, \quad 1? k ? n. \tag{4}$$

The family of functions of the kind ? : ? ^{3} ? ? employed in this approach is radial and can be expressed as

$$? (x) ? ?(\|x\|_2), \quad x ? ? ^3 \tag{5}$$

with ? : ? $_{?0}$? ? and the Euclidean norm is computed in ? ^{3}.

The interpolation function obtained in this way has the limitation of not including linear transformations, this limitation can be overcome, however, by adding a polynomial term [13] from space $?_m^3$ of maximum order *m* that generalizes the expression (2)

$$F(x) ? \sum_{j?1}^{n} w_j ? (x ? x_j) ? \sum_{l?1}^{M} v_l p_l(x) , \tag{6}$$

with $M = dim\ ?_m^3$, $v_1,\ldots,v_M$? ? ^{3} and $p_1,\ldots,p_M$ polynomials in $?_m^3$. Introducing the term

$$\sum_{i?1}^{n} w_i p_l(x_i) ? 0 \qquad 1 ? l ? M \tag{7}$$

The system of equations can be expressed as

$$\begin{matrix} ?A & P^T ? \\ ?P & 0 ? \end{matrix} \begin{matrix} ?w? \\ ?v? \end{matrix} = \begin{matrix} ?y? \\ ?0? \end{matrix} \tag{8}$$

whose solution exists and is unique [16].

It must be noted that the proposed method works correctly also in case of subsets of feature points and it is therefore compliant with the specifications of the Calibration Facial Animation Object Profile which allows for the possibility of transmitting only a subset of the calibration feature points.

The radial basis functions cited in literature are diverse, among the many types proposed so far are:

?? Linear
?? Cubic
?? Thin-plate
?? Gaussian
?? Multi-quadric

For this application, it has been considered appropriate to choose, among the multitude of RBFs, those with the property of being monotone decreasing to zero for increasing values of the radius like Gaussian functions and inverse multi-quadrics. Disadvantages of such functions are mainly due to the global effect produced by the interpolation function that is obtained. Unfortunately, despite the peculiarity just put in evidence, experimental results show an excessive interaction between the various RBF which contribute to generate the interpolation function. Calibration is obtained on

the hypothesis that each feature point (on which RBF are centered) influences a limited region of wire-frame. To make an example, it is quite reasonable to assume that the feature points of the mouth do not interact with the characteristic points of ears and of the upper part of the head.
A particular kind of RBF called Compactly Supported (RBFCS), whose properties are described and discussed in [13,12], have been considered. This type of RBF is widely used in a variety of applications because of the significant computational advantages they offer and the good properties they exhibit like the locality of their domain, characteristics that in our case is of important for bounding the reshaping action applied on the model. The choice made for defining the RBFCS family suitable for solving this specific interpolation problem, was based on empirical considerations as a trade off between computational complexity and subjective quality. In particular, the structure of the employed RBFCS is the following:

$$? _{3,0} ? (1 ? r)_{?}^{7}(5 ? 35r ? 105r^2 ? 147r^3 ? 101r^4 ? 35r^5 ? 5r^6) ? C^6$$
$$? _{3,1} ? (1 ? r)_{?}^{6}(6 ? 36r ? 82r^2 ? 72r^3 ? 30r^4 ? 5r^5) ? C^4$$
$$? _{3,2} ? (1 ? r)_{?}^{5}(8 ? 40r ? 48r^2 ? 25r^3 ? 5r^4) ? C^2$$
$$? _{3,3} ? (1 ? r)_{?}^{4}(16 ? 29r ? 20r^2 ? 5r^3) ? C^0$$

From a careful analysis of the position of the feature points, it can be noticed that some subsets of points (eyes and mouth for instance) are clustered in a small area of the wire-frame, differently than others that are very sparsely distributed. This non-uniformity of feature point distribution makes it complicated to build an interpolation function capable of satisfying both the requirements of precision and smoothness.
The basic idea [14], originated by these requirements, is that of subdividing the ensemble of feature points into a number of subsets:

$$S_0 ? S_1 ... ? S_t ? S ,$$

such that, in each of them, data are distributed in an as uniform fashion as possible, at least for the small ones.
The calibration process consists of doing t levels of interpolation. At the *j-th* level, the residual of the previous level is interpolated by means of the RBFCS $? _j$ of support $? _j$.
After t steps, the resulting interpolation function is:

$$F :? \sum_{k?0}^{t} F_k \qquad (9)$$

In order to preserve the global surface smoothness, low-order RBF are employed for the first iterations, while high-order RBF are applied during the later iterations to add local somatic details.

In the following pictures (Figures 14, 15 and 16) some results are presented showing the effectiveness of the proposed algorithm in reshaping the face models Mike and Oscar by means of feature points extracted from “Cyrano”.

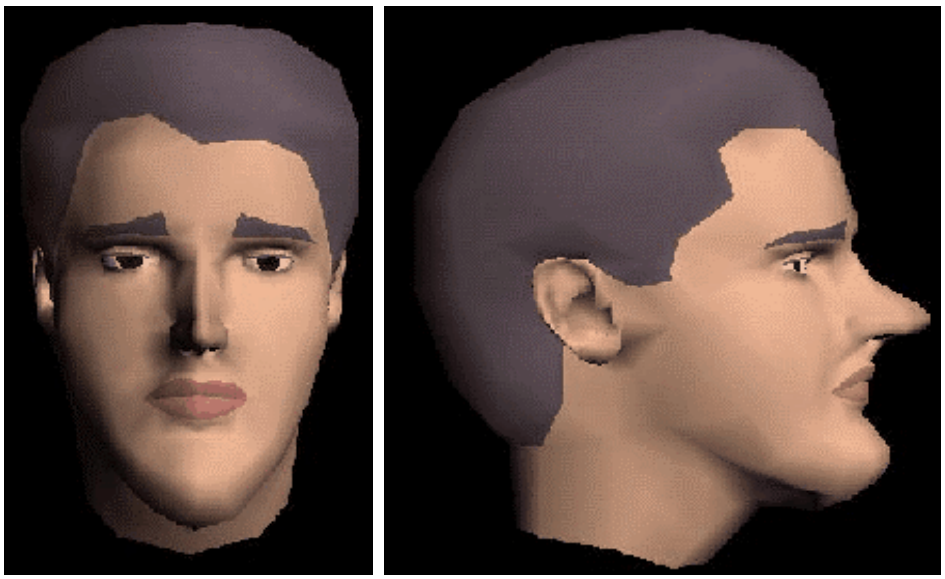


Figure 14: Front and side view of Cyrano (target model taken from [17]).

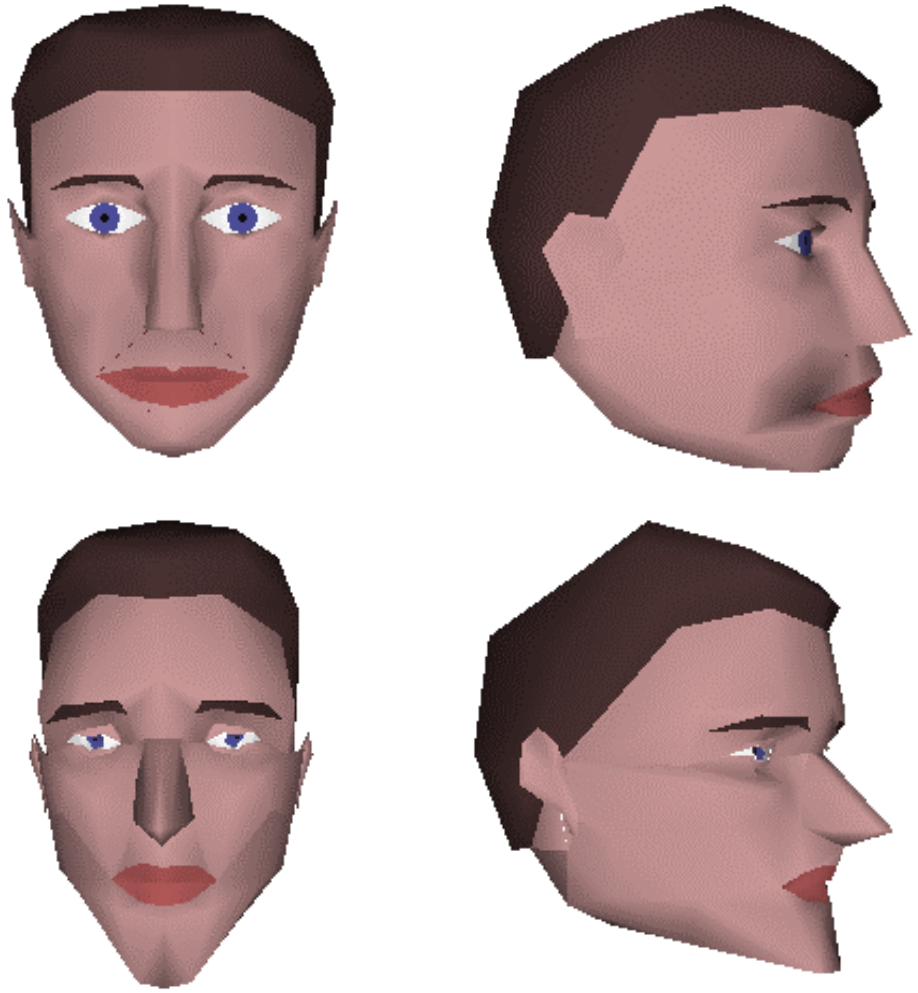


Figure 15: model Mike original (top) and reshaped on “Cyrano” (bottom).

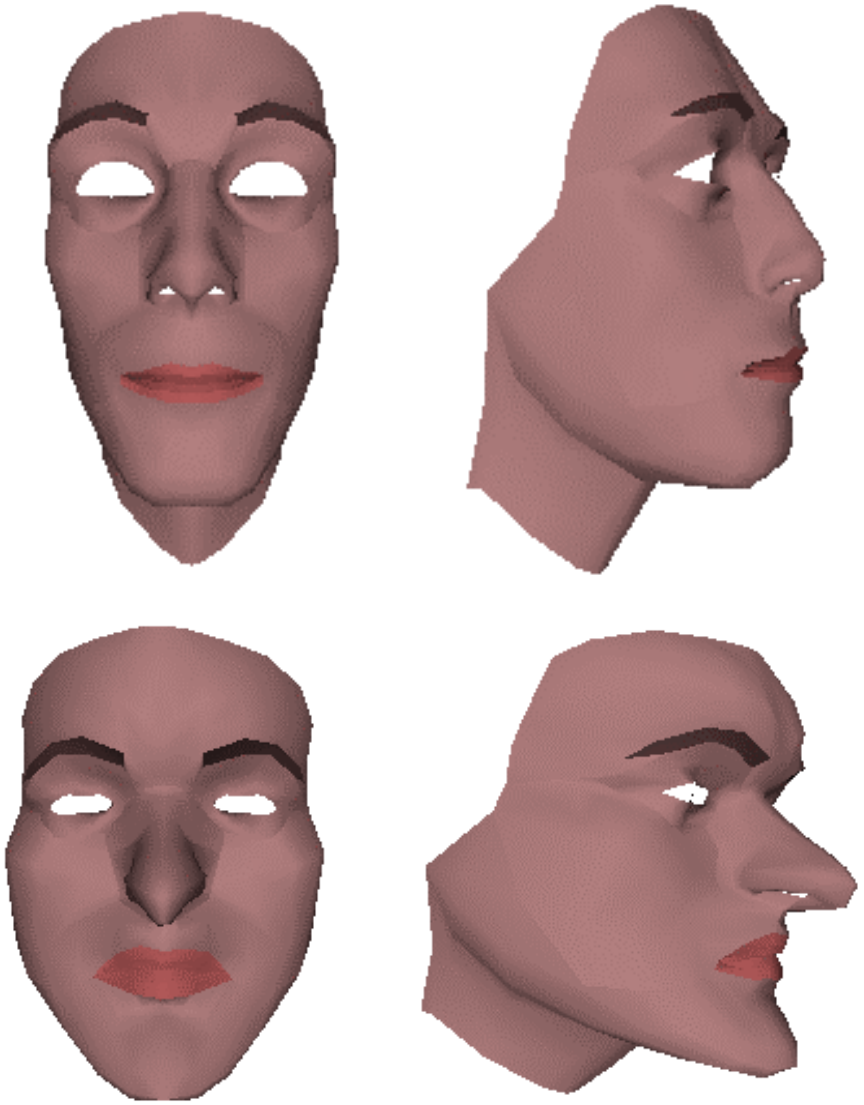


Figure 16: model Oscar original (top) and reshaped on “Cyrano” (bottom).

3.2.2 Model calibration with texture

Besides the model calibration, the standard also includes the possibility of managing the texture information and the texture coordinates for each feature point transmitted in order to allow texture adaptation on the model surface.
The experimental tests which have been carried out, two possible kind of texture:

1. Texture acquired through a 3D scanner
2. Texture extracted from a 2D frontal picture of the face

The first problem to be solved for adding the texture information to the calibrated model, is that of mapping the complete set of 3D model vertices into the 2D-texture domain. The MPEG-4 specifications define the texture coordinates only for the feature points while the decoder mu st compute also the coordinates of the other wire-frame vertices.

To implement this operation a two-step algorithm has been adopted:

?? 2D Projection of the calibrated model into the texture domain
?? Mapping of the 2D projection of the feature points on the texture coordinates

Figure 17: Examples of texture extracted from a 3D scanner (left) and from a 2D picture (right).

The kind of projection that is employed is evidently dependent on the specific kind of texture available.

In the first case, a cylindrical projection is employed from the 3D space on the plane (u, v) by means of the following equations:

$$u = \arctan(x/z)$$
$$v = y$$

In the second case, a planar projection is employed of the kind:

$$u = x$$
$$v = y$$

The same algorithm based on RBF already used for the calibration of the 3D model is used for placing the texture map. The 2D projections of the feature points are forced to coincide with the texture coordinates of the same feature points. In this case, instead of using a multilevel reshaping algorithm, the data has been processed one shot. Clearly, in this case RBF in $?^2$ are used.

3.2.3 Examples of model calibration with texture

An example of the obtained results is reported in the following Figures 18 and 19 showing the two phases of the calibration process, the first being the use of only the feature points and the second being the mapping of the texture information on the reshaped model.

To demonstrate the capabilities of FAE, we present some examples of the results that can be obtained by first applying the calibration process followed by the animation (texture mapping) process. Figures 19 and 20 show two synthesized images obtained by reshaping the model Oscar after its calibration with the FDP of Claude.

Figure 18: (Top) Frontal and side views of the texture calibration target “Claude” [17]; (center) model “Oscar” reshaped with the feature points of “Claude”; (bottom) model “Oscar” reshaped with feature points and with the texture of “Claude”.

Figure 19: Example of model animation obtained by calibrating the model Oscar by means of the FDP of Claude. Facial expression of "anger" without the texture information (top) and with the texture information (bottom), extracted from the expression.fap sequence.

4. Applications

The availability of such a flexible system for the calibration and animation of generic head models through the MPEG-4 parameters offers significant opportunities for designing a wide variety of new services and applications. Besides enriching conventional multimedia products in computer gaming, virtual character animation and augmented reality, very promising applications are foreseen in advanced man-machine interface design, virtual studio design for movie production, teleteaching, telepresence and in videotelephony.

The authors plan to address specific environments related to the use of advanced multimedia technologies in Science Communication, Cultural Heritage, Theater and Arts. In particular, a feasibility study is in progress to implement an interactive system based on a "talking head" to be used by the Aquarium of Genova (the largest in Europe) for guiding visitors to specific sections of the exposition, similar investigation is in progress in cooperation with the acting company

"Teatro dell'Archivolto" of the historical Genoese theater "Gustavo Modena" for writing novel pieces of unconventional theater with natural actors playing on the stage together with virtual actors "living" on a screen.

Significant efforts are currently underway for using the talking head by interactive CD-ROMs for applications like story telling, with the possibility of reshaping the model to the face of known persons like actors, singers, athlets or cartoon characters, for speech rehabilitation of deaf and for teaching foreign languages.

On the other hand, attention is being paid to the development of robust tools for automatically capturing a human face parameters from audio, video and text, possibly in real-time, in order to encode the various facial information (texture, FDP and FAP) directly from natural multimodal sources. Progress in this direction have been made recently, based on the use of Time-Delay Neural Networks for the direct mapping from speech to FAP [18], with reference to FAP1 representing visemes.

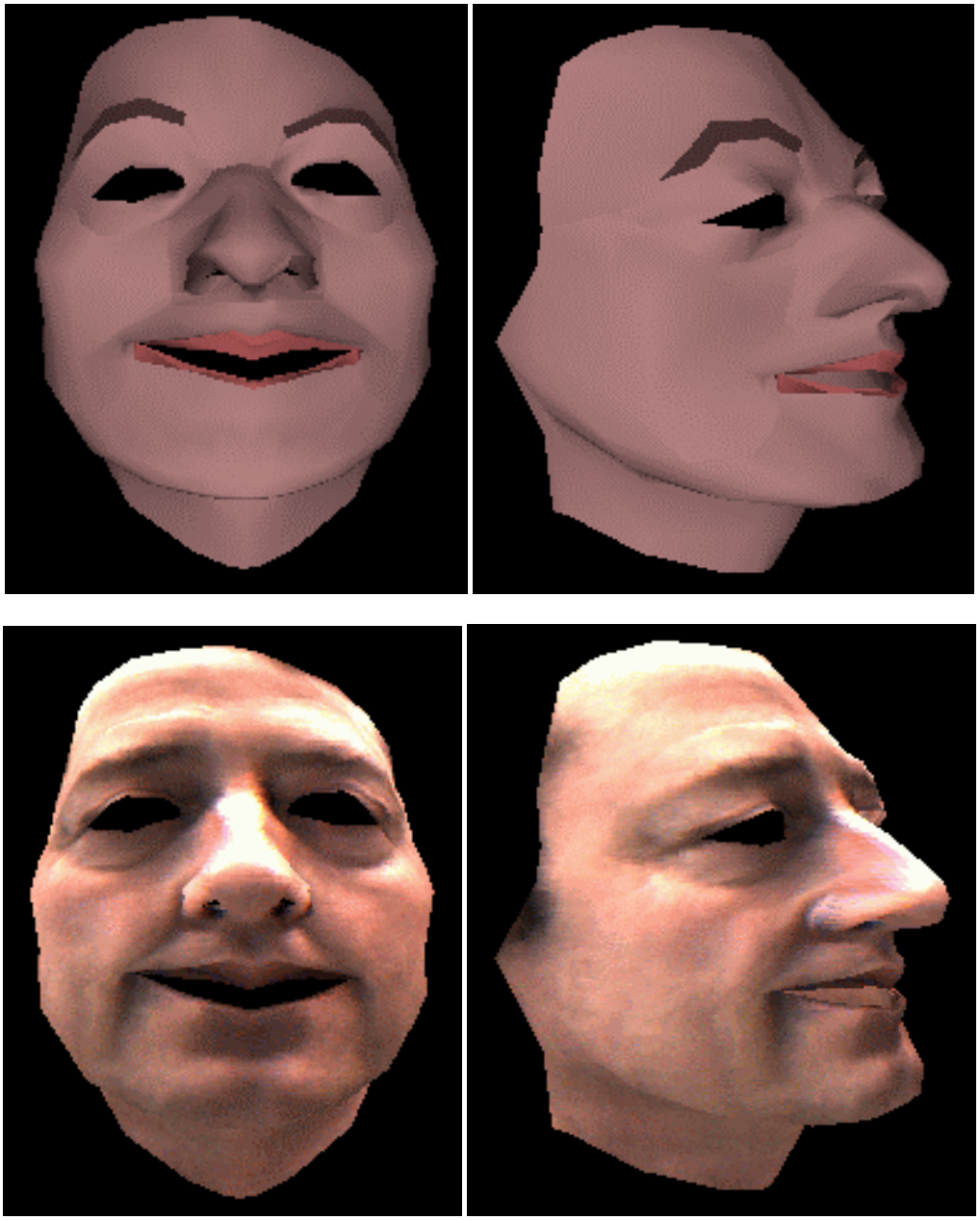


Figure 20: Example of model animation obtained by calibrating the model Oscar by means of the FDP of Claude. Facial expression of "smile" without the texture information (top) and with the texture information (bottom), extracted from the Marco20.fap sequence.

5. Conclusions

A particular implementation of a Face Animation Decoder compliant with MPEG-4 has been described, based on a proprietary animation software called FAE, capable of animating a generic facial wire-frame by providing the usual geometric parameters together with some semantic information on the wire-frame.
FAE, starting from this high-level information, automatically generates the animation rules suited to that particular wire-frame and subsequently animates the model by means of a stream of FAP. Therefore, FAE is compliant with the specifications of the Simple Facial Animation Object Profile. It has the advantage of being able to easily handle different kinds of models without any dependence on the specific model.
Besides this, FAE includes also a calibration module capable of reshaping and animating the proprietary model by means of FDP information (feature points, texture and texture coordinates).
This peculiarity makes it suited for the implementation of the Calibration Facial Animation Object Profile according to the specifications defined at the San Jose meeting. Complete compatibility with the Calibration profile will be reached with the implementation also of FIT.

Many aspects are still to be improved. Implementation of the movements affecting the ears, the nose and the tongue is still missing. The pre-defined movements should be improved further by experimenting with new criteria for the computation of the weights to make the face movements more realistic. Finally the properties of RBF should be investigated more deeply, defining sub-sets of feature points for the multilevel approach and varying the size of the support and the typology of the RBF, depending on the specific feature points on which they are centered.

Bibliography

[1] MPEG Systems, *"Text for CD 14496-1 Systems"*, ISO/IEC JTC1/SC29/WG11/N1901, 1997.
[2] MPEG Video, *"Text for CD 14496-2 Video"*, ISO/IEC JTC1/SC29/WG11/N1902, 1997.
[3] MPEG SNHC, *"Initial thoughts on SNHC visual object profiling and levels"*, ISO/IEC JTC1/SC29/WG11/N1972, 1997.
[4] MPEG Systems, *"Study of Systems CD"*, ISO/IEC JTC1/SC29/WG11/N2043, 1998.
[5] MPEG Video and SNHC, *"Study of CD 14496-2 (Visual)"*, ISO/IEC JTC1/SC29/WG11/N2072, 1998.
[6] C. Braccini, S. Curinga, A. Grattarola, F. Lavagetto, *"Muscle Modeling for Facial Animation in Videophone Coding"*, IEEE International Workshop on Robot and Human Communication, pp. 222-226, 1992.
[7] P. Eckman e W.E. Friesen, *"Facial Action Coding System"*, Consulting Pychologist Press, Palo Alto (Califomia), 1977.
[8] R. Koenen, F. Pereira, L. Chiariglione, *"MPEG-4: Context and Objectives"*, Image Communication Journal, May 1997.
[9] A. Morelli e G. Morelli, *"Anatomia per gli Artisti"*, Fratelli Lega Editori, Faenza (Italy), 1987
[10] F. I. Parke, *"Parametrized Models for Facial Animation"*, IEEE Computer Graphics Applications, 2(9):61-68, Novemer 1982.
[11] F. I. Parke, K. Waters, *"Computer Facial Animation"*, A K Peters Ltd, MA, 1996.
[12] O. Soligon, A. Le Mehaute, C. Roux, *"Facial Expression simulation with RBF"*, Internatonal Workshop on SNHC and 3D Imaging, Rhodos (Greece), pp. 233-236, September 1997.
[13] R. Schaback, *"Creating surfaces from scattered data using radial basis functions"*, Mathematical methods in CAGD III, M.Daelhen, T.Lyche, and L. Shumacker (eds.), pp. 1-21, 1995.
[14] F.J. Narcowich, R. Schaback, J.D.Ward, *"Multilevel interpolation and approximation"*, Int. Journal on Applied and Computational Harmonic Analysis, Academic Press (in press).
[15] P. Alfeld, *"Scattered data interpolation in three or more variables"*, Mathematical methods in CAGD III, M.Daelhen, T.Lyche, and L. Shumacker (eds.), Academic Press, 1989.
[16] C.A. Micchelli, *"Interpolation of scatterd data: distance matrices and conditionally positive definite functions"*, Constr. Approx., 2:11-22, 1986.
[17] FBA Test Data Set: *"http://www-dsp.com.dist.unige.it/snhc/fba_ce/"*.
[18] F.Lavagetto, *"Converting speech into lip movements: A multimedia telephone for hard of hearing people"*, IEEE Trans. on Rehabilitation EngineeringVol.3, n.1, pp. 90-102, 1995.