The Facial Animation Engine: Toward a High-Level Interface for the Design of MPEG-4 Compliant Animated Faces

Fabio Lavagetto and Roberto Pockaj

***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 wireframe 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 facial animation parameters. If the calibration parameters (feature points and texture) are available, the system is able to exploit this information for suitably modifying the geometry of the wireframe 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 this paper by means of figures 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.

***Index Terms*—Facial modeling and animation, graphic interfaces, multimedia management, synthetic video.**

I. Introduction

MPEG-4 activities began 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 applications in digital TV). MPEG-4 has as its main mandate the definition of an efficient system for encoding audiovisual objects (AVO's), either natural or synthetic [8]. This means that, toward the revolutionary objective of going beyond the conventional concept of the audiovisual scene as composed rigidly of a sequence of rectangular video frames with associated audio, MPEG-4 is suitably structured to manage any kind of AVO, with the capability to compose them variously in two-dimensional (2-D) and three-dimensional (3-D) scenes. The heart of MPEG-4, therefore, is composed of the architecture of systems, responsible for the definition of the respective position of objects composing the scene; their behavior; and their interactions. This information is described through a hierarchical description within the so-called "scene graph." A typical, frequently used scheme of an MPEG-4 scene is represented in Fig. 1.

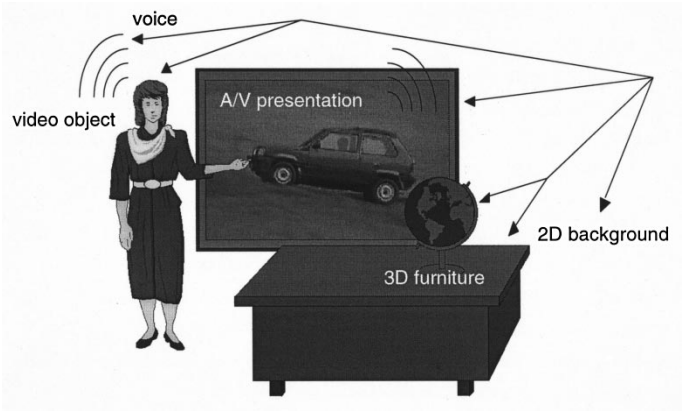


Fig. 1. Typical MPEG-4 scene with its associated "scene graph."

According to the scheme sketched in Fig. 2, the various audiovisual 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 audiovisual 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 audiovisual 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 3-D wireframes 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 their dynamic behavior (smoothness in rendering facial movements and realism in performing facial expressions).

Manuscript received October 1, 1997; revised November 1, 1998. This paper was recommended by Guest Editors H. H. Chen, T. Ebrahimi, G. Rajan, C. Horne, P. K. Doenges, and L. Chiariglione.

The authors are with the Department of Communications, Computer and System Science (DIST), University of Genova, Genova 16145 Italy (e-mail: fabio@dist.unige.it; pok@dist.unige.it).

Publisher Item Identifier S 1051-8215(99)02331-9.

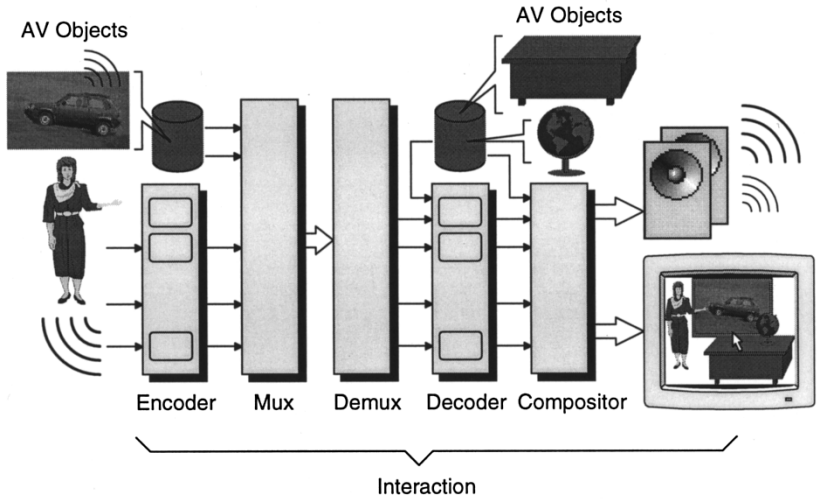


Fig. 2. Scheme of the MPEG-4 system.

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 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.

Facial animation parameters (FAP's) 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 other words, FAP represents the proper animation parameter stream.

Facial definition parameters (FDP's) are responsible for defining the appearance of the face. These parameters can be used either to modify (though with various levels of fidelity) the shape and appearance 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's. FDP's, on the contrary, are typically employed only when a new session is started.

A FAP interpolation table (FIT) is used to reduce the FAP bit rate. 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 a "tool," it is necessary to introduce the definition of an "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 in Section II-B, represent a set of key-points on the face that are used by the decoder to adapt its generic model to a specific 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's must be used, included those responsible for downloading the model from the bitstream. By using a specific combination of FDP's (which includes the use of a 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 (CD) [5] and to the Systems CD [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

Fig. 3. The neutral face.

this novel technology into consolidated multimedia products and services. As an example, challenging opportunities 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 Forty-Fourth MPEG-4 meeting, held in Dublin, Ireland, in July 1998, the facial model described in this paper was given to the MPEG implementation studies group (ISG) for testing candidate algorithms for evaluating the complexity introduced by facial rendering.[1]

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 II, together with a description of the standardized facial parameters. A detailed description of the facial model implemented by the authors is provided in Section III, while a few preliminary examples of technology transfer and its application are described in Section IV. Final conclusions are drawn in Section V.

II. Facial Animation Semantics

A. 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 introduce the concept of neutral face as defined in [2], corresponding to the face posture represented in Fig. 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 muscles 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 as the lip corners.

[1] Sample images and demo movies of the facial model will be available at http://www-dsp.com.dist.unige.it.

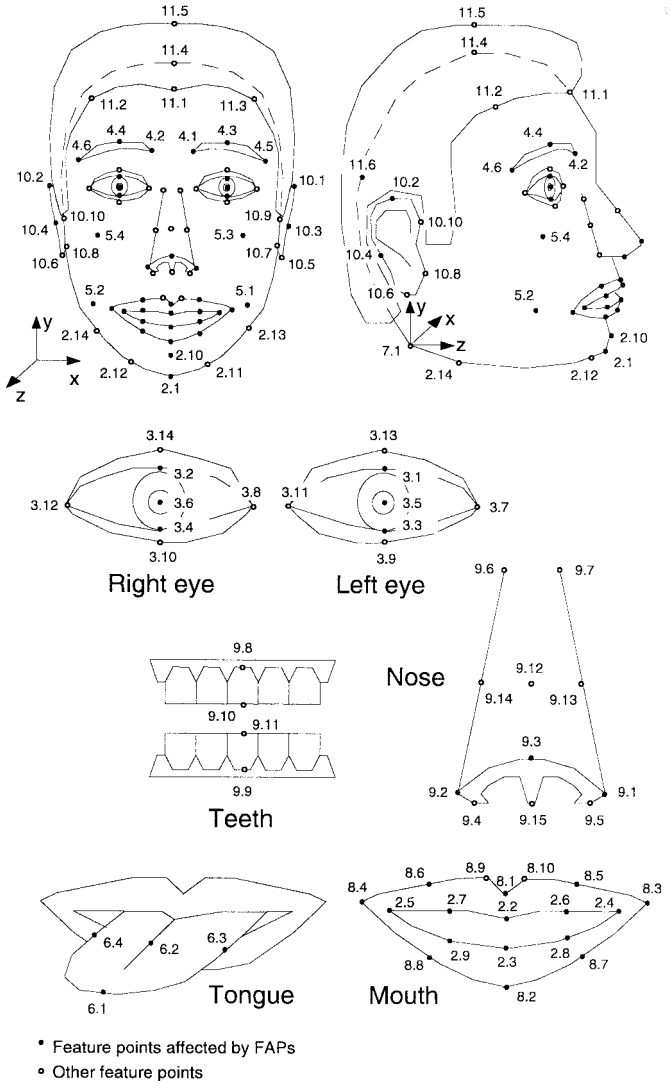


Fig. 4. Feature points.

- 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 the upper and lower teeth.

B. 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 Fig. 4. Table I lists the characteristics, the location, and some recommended constraints for the feature points in the eyebrow region.

C. FAP's and Their Measurement Units (FAPU's)

While all the feature points contribute to defining the appearing of the face in the calibration facial animation object

TABLE I
FEATURE POINTS DESCRIPTION TABLE

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 II
FACIAL ANIMATION PARAMETERS DESCRIPTION TABLE

#	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
...	...	...	...	...	...	...	...	...

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, an 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 in Table II. In each 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 are monodirectional or bidirectional, an indication of the positive direction of the motion, the feature point affected by the

TABLE III
FACIAL ANIMATION PARAMETERS UNITS DESCRIPTION TABLE. SYMBOLS AND FORMULAS IN THE LEFT-HAND COLUMN REFER TO THE FACIAL FEATURE POINTS SHOWN IN FIG. 4. EXTENSIONS .x AND .y STAND FOR THE HORIZONTAL AND VERTICAL COORDINATE OF THE FEATURE POINT, RESPECTIVELY

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

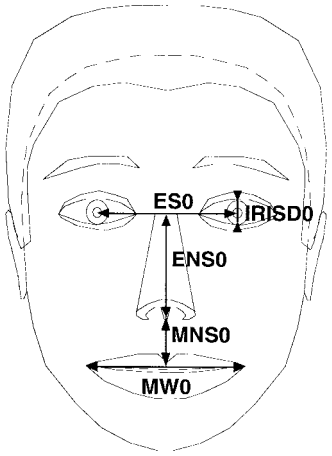


Fig. 5. Facial animation parameter units.

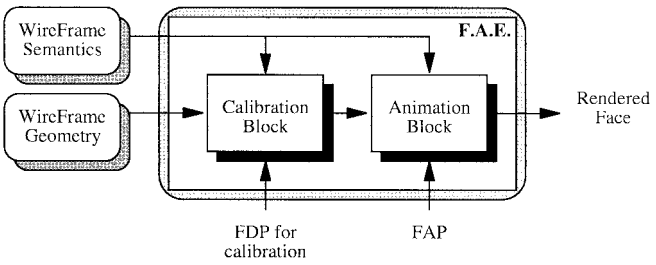


Fig. 6. Block diagram of the face animation engine.

FAP (by identifying the group and the subgroup to which it belongs), and the quantization step.

The magnitude of the displacement is expressed by means of specific measure facial animation parameter units (FAPU's). Each FAPU represents a fraction of a key-distance on the face (as an example, 1/1024 of the mouth width), except for FAPU's 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 III and in Fig. 5.

The method for interpreting the value of FAP is, therefore, quite simple. After decoding an FAP, its value is converted in the measurement 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, the computed displacement. Since an FAP defines only how to compute one 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 that compose the wireframe 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 wireframe by exploiting only the knowledge of its topology and of the semantics of its vertices.

III. THE FACIAL ANIMATION ENGINE

The core mechanisms responsible for the facial animation are based on software procedures that compose a high-level interface for the implementation of 3-D 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 wireframe 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 the wireframe 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 the proprietary model reshaped according to the calibration FDP. A high-level description of the FAE is shown in Fig. 6.

A. 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 net of polygons. It contains the coordinates of the vertices and the topology of the wireframe 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 wireframe vertices that 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 wireframe vertices, so that these semantic definitions appear to be immediately comprehensible starting with any generic wireframe.

Besides the list of points that are moved by each specific FAP, the face animation engine operates on a table that associates with each FAP a predefined movement that is applied to the associated vertices. It is apparent that different FAP's can use the same predefined movement, so that it is reasonable to assume significantly fewer movements than FAP's. Some examples of predefined 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 predefined movements, the animation engine automatically computes the animation rules.

To describe this procedure, we introduce an example. Let us consider, among the many movements that can affect the left eyebrow, the one that is controlled by FAP31 (raise_l_i_eyebrow), which controls the vertical displacement of the eyebrow internal region.

From the semantics associated to the wireframe, 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 predefined movement associated to FAP31 is the vertical translation along $Y + Z$.

In Fig. 7, the feature point associated to FAP31 is identified on the wireframe region around the left eyebrow, together with the vertices that are affected by the predefined movement associated with FAP31 and the bounding points (vertices) that limit the domain for the effects produced by the movement itself.

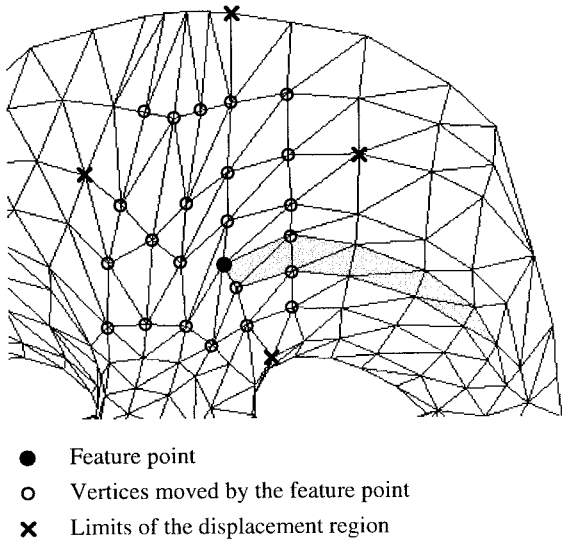


Fig. 7. Information necessary to compute the animation rule controlled by FAP31.

For each vertex that 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 Section III-A1.

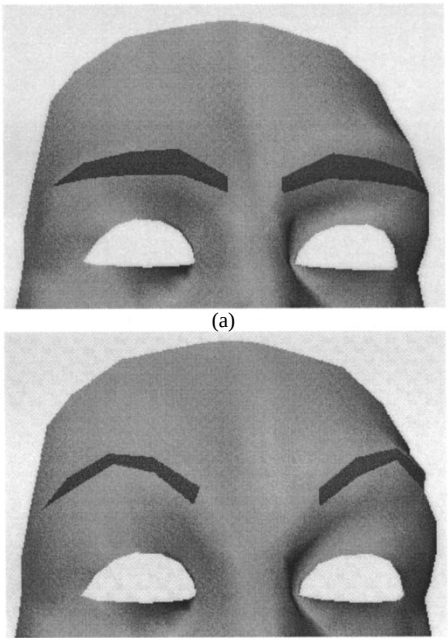
Once a weight is associated with each vertex, the specific predefined movement is applied accordingly. In the case of FAP31, as said before, it is a vertical translation along $Y + Z$. However, along with its vertical movement, the eyebrow slides smoothly upwards on the skull, so that the z-coordinate 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.

Figs. 8 and 9 show how, adopting the criteria described above, 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 nonsymmetrical movements.

It must be noted how the displacement domain of feature points can overlap one another (i.e., a vertex of the wireframe not coinciding with a feature point can be affected by the movements associated with different feature points). Fig. 9 shows the three displacement domains associated with the three FAP's operating on the left eyebrow.

1) Description of the Predefined Movements:

a) Translation parallel to X, Y, or Z: In the 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 wireframe. For the computation of the weights, in particular, it is necessary to know the vertex identified as a feature point, the vertices affected by the predefined movement, and the vertices that define the displacement domain.



(a) (b)

Fig. 8. (a) Eyebrow squeezing and (b) eyebrow raising; automatically generated animation rules with the "$Y + Z$ translation" predefined animation.

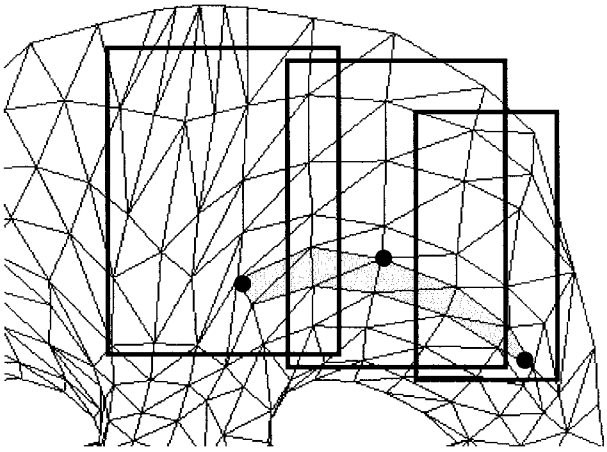


Fig. 9. Displacement domains for FAP31, FAP33, and FAP35.

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

$$W_i = \frac{\Delta x_i}{\Delta X} \cdot \frac{\Delta y_i}{\Delta Y}$$

where, with reference to Fig. 10, P_i represents the ith 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 that limit the displacement domain.

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

$$\textit{Displacement\_of\_feature\_point} * W_i.$$

b) 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 was described for the translations

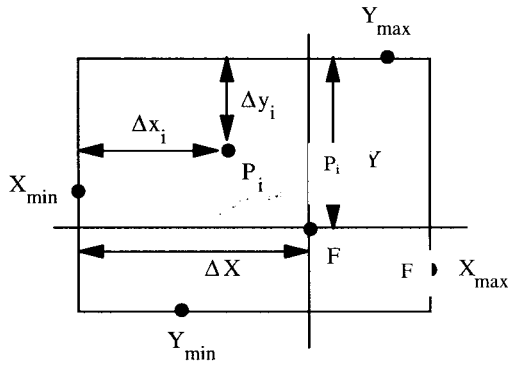


Fig. 10. Algorithm for computing the weights associated to vertices.

along one single axis except in this case, also the z-coordinate is affected by the movement.

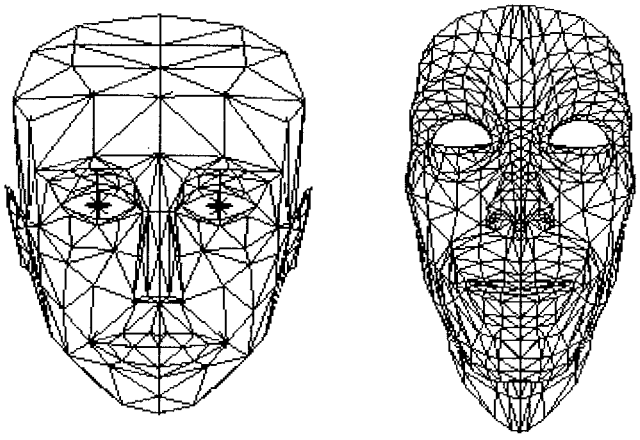
By sampling the neutral face, an approximation of the 3-D geometry of the head is obtained, which is then used to constrain the trajectories of the feature points and of the related vertices to lie on its surface. Exploiting this 3-D *a priori* knowledge, it is possible to apply vertex translations on 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.

c) Proper rotation around an axis parallel to X, Y, or Z: From the semantic information about the wireframe, 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 movement reproduces rigid rotations of the entire wireframe or only of parts of it.

This movement is applied both for those FAP's that 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.

d) Weighted X rotation: This kind of movement determines the rotation of a set of vertices around the coordinated axis X of an angle that varies vertex by vertex and is expressed as a function of the vertices' linear displacement along the axis Y. Also in this case, the weights that 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).

2) Examples of Animation: In the following figures, we report some results obtained using the data available in the test data set of the FBA ad hoc group of MPEG-4. The facial animation engine has been employed to animate two different models. "Mike" is a proprietary design of Department of Communications, Computer and Systems Science (DIST), University of Genova, Italy. 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



(a) (b)

Fig. 11. Wireframe of (a) "Mike" and (b) "Oscar."

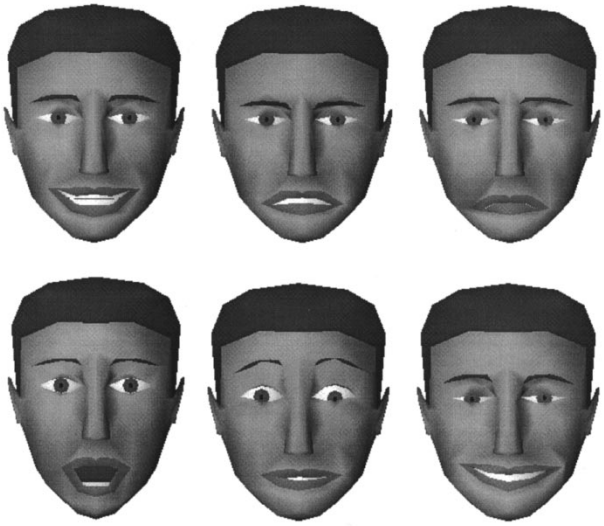


Fig. 12. Facial expression of joy, anger, sadness, and surprise extracted from the file "expressions.fap," and eyebrow raising and smile extracted from the file "Marco20.fap," reproduced on the model Mike.

Parke and Waters [11]; it is more complex but still incomplete in some parts. As shown in Fig. 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 depending on 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 Fig. 11, the wireframe 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.

Figs. 12 and 13 show facial expressions, eyebrow raising, and smile extracted from the file "Marco20.fap" reproduced on Mike and Oscar, respectively.

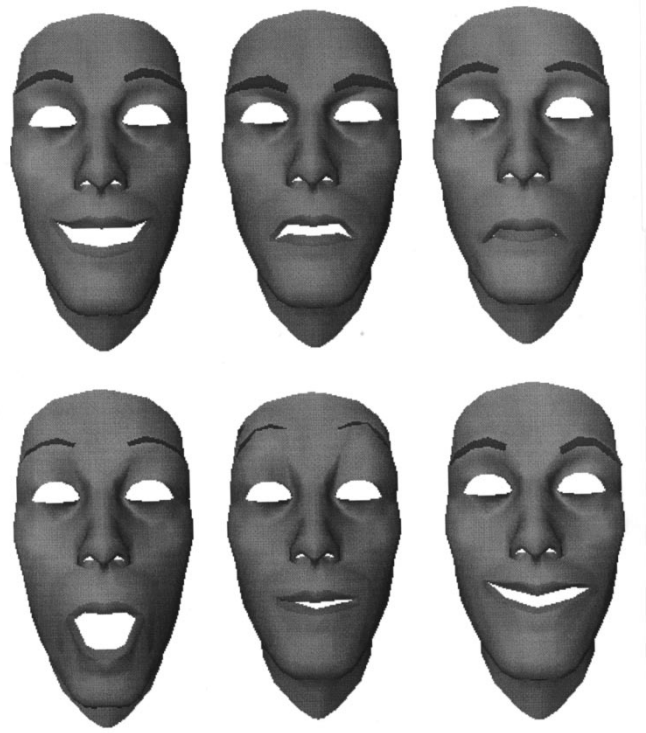


Fig. 13. Facial expression of joy, anger, sadness, and surprise extracted from the file "expressions.fap," and eyebrow raising and smile extracted from the file "Marco20.fap," reproduced on the model Oscar.

B. 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's).

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 wireframe, which, depending on its complexity, 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 N points $S \in \Re^3$, including a subset $X = \{x_1, \cdots, x_N\} \subset S$, representing the ensemble of feature

points. Let us associate with X a corresponding set $Y = \{y_1, \cdots, y_n\} \in \Re^3$ indicating the subset of calibration feature points transmitted by the encoder.

The problem to solve is to define an interpolation function F: $\Re^3 \rightarrow \Re^3$ such that

$$F(x_i) = y_i, \qquad 1 \leq i \leq n \tag{1}$$

and that produces realistic results.

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

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

where $w_j \in \Re^3$ represents the weight associated to the jth function and the function Φ: $\Re^3 \rightarrow \Re$ is positive definite on $\Re^3$ in the sense that for any finite set of points $X = \{x_1, \cdots, x_n\} \in \Re^3$, the matrix

$$A = (\Phi(x_k - x_j))_{1 \leq j,\, k \leq n} \tag{3}$$

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

$$F(x_k) = \sum_{j=1}^{n} w_j \Phi(x_k - x_j) = y_k, \qquad 1 \leq k \leq n. \tag{4}$$

The family of functions of the kind Φ: $\Re^3 \rightarrow \Re$ employed in this approach is radial and can be expressed as

$$\Phi(x) = \phi(||x||_2), \qquad x \in \Re^3 \tag{5}$$

with ϕ: $\Re_{\geq 0} \rightarrow \Re$, and the Euclidean norm is computed in $\Re^3$.

With this interpolation function, however, linear transformations between the proprietary and the target feature points are approximated through nonlinear deformations. This limitation can be overcome by adding a polynomial term [13] of order $m = 1$ from space $\mathbf{P}_m^3$, responsible for performing rigid roto-translation and scale compensation of the proprietary model, while fine adaptation is due to the RBF's contribution. The generalized expression of (2) becomes

$$F(x) = \sum_{j=1}^{n} w_j \Phi(x - x_j) + \sum_{l=1}^{M} v_l p_l(x) \tag{6}$$

with $M = \dim \mathbf{P}_m^3$, $v_1, \cdots, v_M \in \Re^3$, and $p_1, \cdots, p_M$ polynomials in $\mathbf{P}_m^3$. Introducing the term

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

the system of equations can be expressed as

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

whose solution exists and is unique [16].

It must be noted that the proposed method works correctly also in the 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;
- multiquadric.

For this application, it has been considered appropriate to choose, among the multitude of RBF's, those with the property of being monotone decreasing to zero for increasing values of the radius like Gaussian functions and inverse multiquadrics. 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 among the various RBF's that contribute to generate the interpolation function. Calibration is obtained on the hypothesis that each feature point (on which RBF's are centered) influences a limited region of wireframe. As 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 [12] and [13], has been considered. These RBF's are 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, a characteristic that in our case is of importance 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 tradeoff between computational complexity and subjective quality. In particular, the structure of the employed RBFCS is the following:

$$\begin{aligned}
\Phi_{3,0} &= (1-r)_+^7 \left(5 + 35r + 105r^2 + 147r^3 \right. \\
&\qquad \left. + 101r^4 + 35r^5 + 5r^6\right) \in C^6 \\
\Phi_{3,1} &= (1-r)_+^6 \left(6 + 36r + 82r^2 + 72r^3 \right. \\
&\qquad \left. + 30r^4 + 5r^5\right) \in C^4 \\
\Phi_{3,2} &= (1-r)_+^5 \left(8 + 40r + 48r^2 + 25r^3 + 5r^4\right) \in C^2 \\
\Phi_{3,3} &= (1-r)_+^4 \left(16 + 29r + 20r^2 + 5r^3\right) \in C^0.
\end{aligned}$$

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 wireframe, differently than others that are very sparsely distributed. This nonuniformity 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 $X_0 \subset X_1 \cdots \subset X_t = X$ such that, in each of them, data are distributed in as uniform a fashion as possible, at least for the small ones.

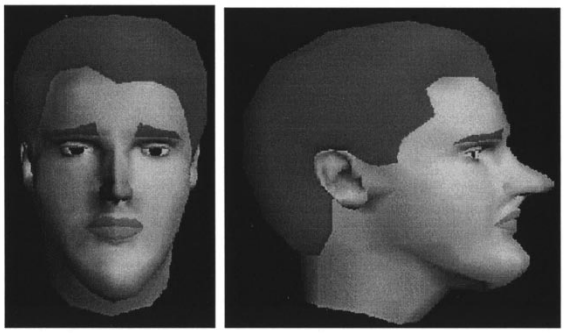
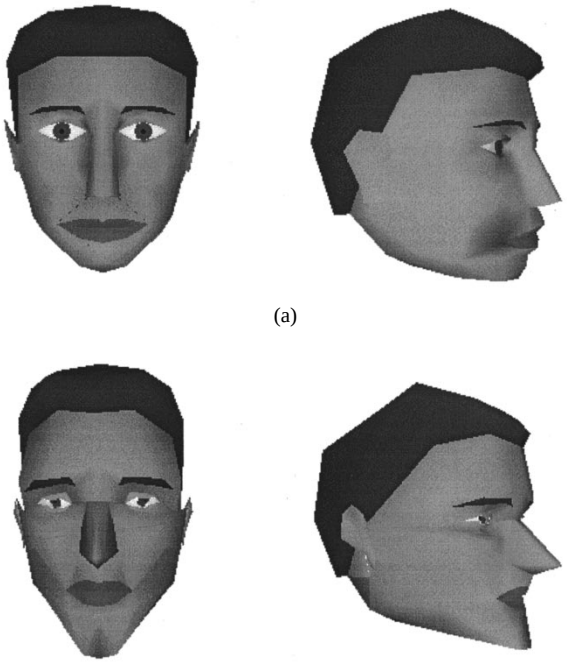


Fig. 14. Front and side view of Cyrano (target model taken from [17]).



(a) (b)

Fig. 15. (a) Model Mike original and (b) reshaped on Cyrano.

The calibration process consists of doing t levels of interpolation. At the jth level, the residual of the previous level is interpolated by means of the RBFCS ϕ_j of support σ_j.

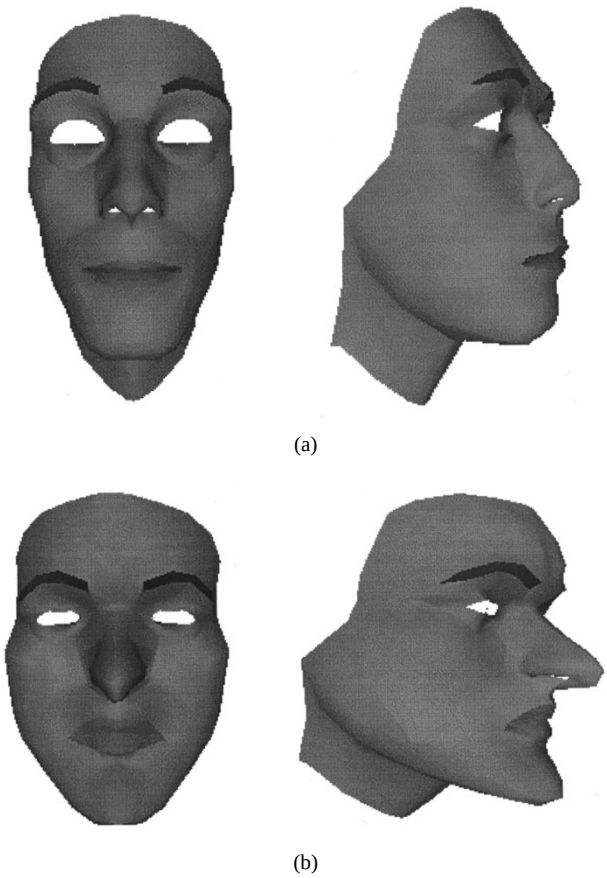
After t steps, the resulting interpolation function is

$$F := \sum_{k=0}^{t} F_k. \tag{9}$$

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

In Figs. 14–16, 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."

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



(a) (b)

Fig. 16. (a) Model Oscar original and (b) reshaped on Cyrano.

(a) (b)

Fig. 17. Examples of texture extracted (a) from a 3-D scanner and (b) from a 2-D picture.

on the model surface. The experimental tests that have been carried out have two possible kinds of texture:

1) texture acquired through a 3-D scanner [see Fig. 17(a)];
2) texture extracted from a 2-D frontal picture of the face [see Fig. 17(b)].

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

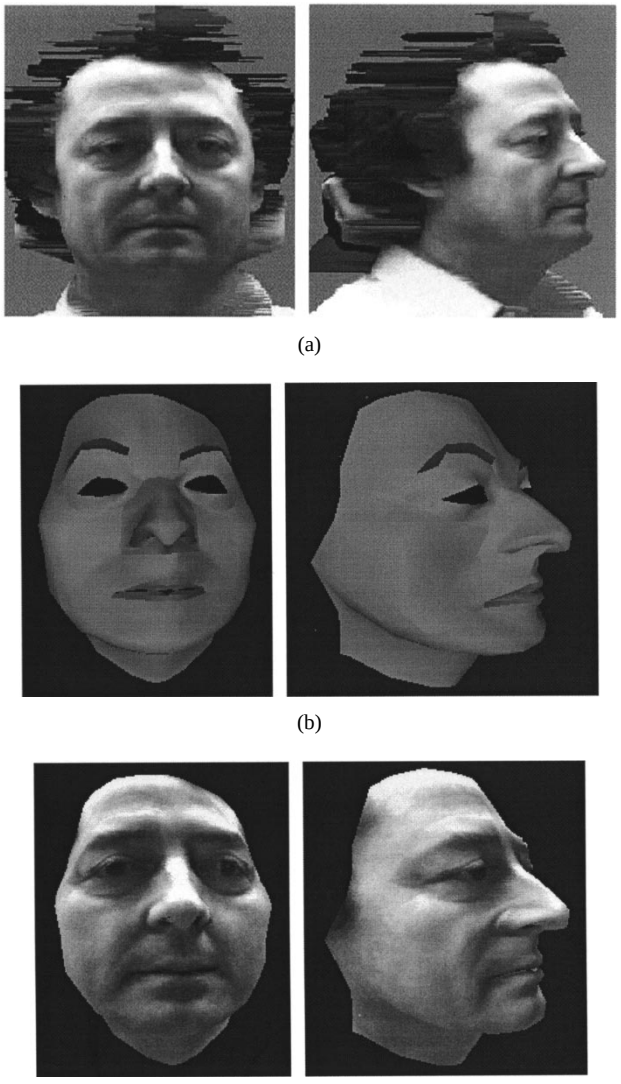


Fig. 18. (a) Frontal and side views of the texture calibration target "Claude" [17], (b) model Oscar reshaped with the feature points of Claude, and (c) model Oscar reshaped with feature points and with the texture of Claude.

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

- 2-D projection of the calibrated model into the texture domain;
- mapping of the 2-D projection of the feature points on the texture coordinates.

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 3-D 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

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

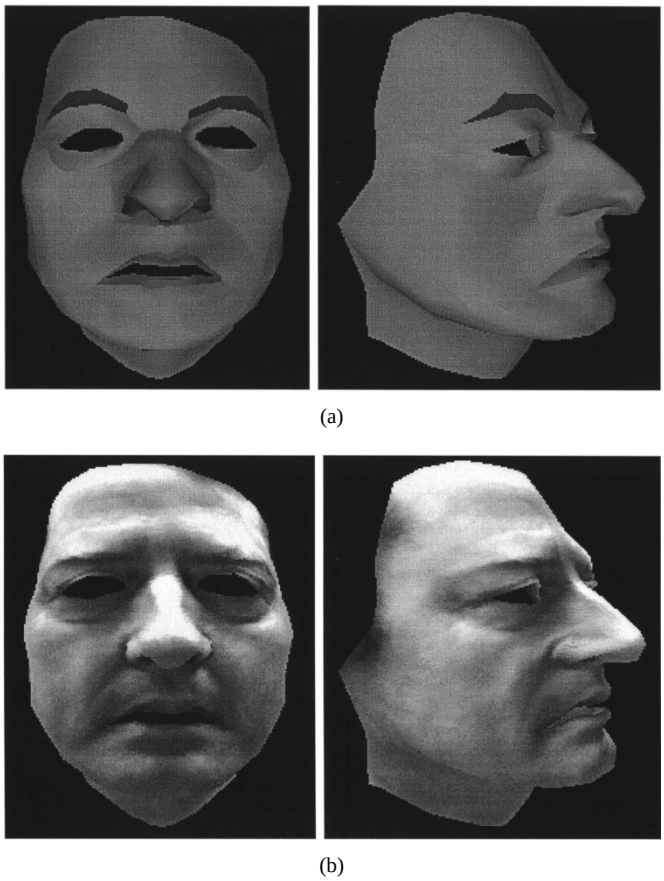


Fig. 19. Example of model animation obtained by calibrating the model Oscar by means of the FDP of Claude. Facial expression of "anger" (a) without the texture information and (b) with the texture information, extracted from the expression.fap sequence.

The same algorithm based on RBF already used for the calibration of the 3-D model is used for placing the texture map. The 2-D 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 have been processed in one shot. Clearly, in this case, RBF's in $\Re^2$ are used.

3) Examples of Model Calibration with Texture: An example of the obtained results is reported in Figs. 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. Figs. 19 and 20 show two synthesized images obtained by reshaping the model Oscar after its calibration with the FDP of Claude.

IV. 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

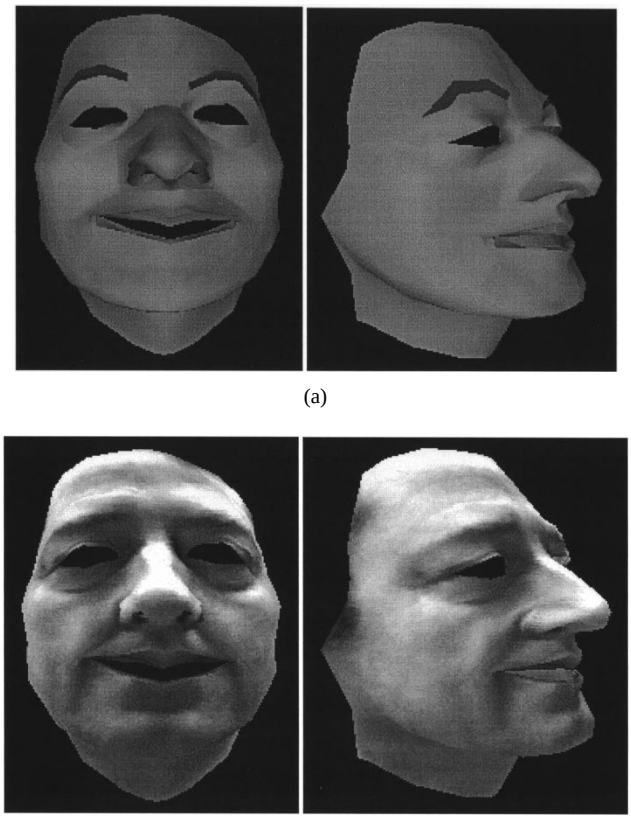


Fig. 20. Example of model animation obtained by calibrating the model Oscar by means of the FDP of Claude. Facial expression of "smile" (a) without the texture information and (b) with the texture information, extracted from the Marco20.fap sequence.

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 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. A similar investigation is in progress in cooperation with the acting company "Teatro dell'Archivolto" of the historical Genovese 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 under way for using the talking head by interactive CD-ROM's for applications like storytelling, with the possibility of reshaping the model to the face of known persons like actors, singers, athletes, or cartoon characters, for speech rehabilitation of the deaf, and for teaching foreign languages.

Also, attention is being paid to the development of robust tools for automatically capturing 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 has 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.

V. Conclusions

A particular implementation of a face animation decoder compliant with MPEG-4 has been described. It is based on a proprietary animation software called FAE, capable of animating a generic facial wireframe by providing the usual geometric parameters together with some semantic information on the wireframe.

FAE, starting from this high-level information, automatically generates the animation rules suited to that particular wireframe and subsequently animates the model by means of a stream of FAP's. 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.

In addition, 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 predefined movements should be improved further by experimenting with new criteria for the computation of the weights to make the face movements more realistic. Last, the properties of RBF should be investigated more deeply, defining subsets of feature points for the multilevel approach and varying the size of the support and the typology of the RBF's, depending on the specific feature points on which they are centered.

References

[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/N1901, 1998.
[6] C. Braccini, S. Curinga, A. Grattarola, and F. Lavagetto, "Muscle modeling for facial animation in videophone coding," in *Proc. IEEE Int. Workshop Robot and Human Communication*, 1992, pp. 222–226.
[7] P. Eckman and W. E. Friesen, *Facial Action Coding System.* Palo Alto, CA: Consulting Psychologist Press, 1977.
[8] R. Koenen, F. Pereira, and L. Chiariglione, "MPEG-4: Context and objectives," *Image Commun. J.*, May 1997.
[9] A. Morelli and G. Morelli, *Anatomia per gli Artisti.* Faenza, Italy: Fratelli Lega Editori, 1987.
[10] F. I. Parke, "Parametrized models for facial animation," *IEEE Computer Graph. Appl. Mag.*, vol. 2, no. 9, pp. 61–68, Nov. 1982.
[11] F. I. Parke and K. Waters, *Computer Facial Animation.* Wellesley, MA: A. K. Peters, 1996.

[12] O. Saligon, A. Le Mehaute, and C. Roux, "Facial expression simulation with RBF," in *Proc. Int. Workshop SNHC and 3D Imaging*, Rhodos, Greece, Sept. 1997, pp. 233–236.
[13] R. Schaback, "Creating surfaces from scattered data using radial basis functions," in *Mathematical Methods in CAGD III*, M. Daelhen, T. Lyche, and L. Shumacker, Eds. New York: Academic, 1995, pp. 1–21.
[14] F. J. Narcowich, R. Schaback, and J. D. Ward, "Multilevel interpolation and approximation," *Int. J. Appl. Computat. Harmonic Anal.*, to be published.
[15] P. Alfeld, "Scattered data interpolation in three or more variables," in *Mathematical Methods in CAGD III*, M. Daelhen, T. Lyche, and L. Shumacker, Eds. New York: Academic Press, 1989.
[16] C. A. Micchelli, "Interpolation of scattered data: Distance matrices and conditionally positive definite functions," *Constr. Approx.*, vol. 2, pp. 11–22, 1986.
[17] FBA Test Data Set. [Online]. Available WWW: 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. Rehab. Eng.*, vol. 3, no. 1, pp. 90–102, 1995.

Fabio Lavagetto was born in Genova, Italy, on June 8, 1962. He received the master's degree in electronic engineering and the Ph.D. degree from the University of Genova in 1987 and 1992, respectively.

In November 1988, he joined Marconi SpA, where he worked on real-time systems design for IR image processing. He then joined the Department of Communication, Computer and System Science (DIST), University of Genova. In 1990, he was a Visiting Researcher at the Visual Communication Department of AT&T Bell Labs. In 1993, he was a Contract Professor at the University of Parma, teaching a class on digital signal processing. Since 1994, he has been a tenured Assistant Professor at the University of Genova teaching a class on radio communication systems. He is responsible for scientific aspects of DIST within several national and European projects. Since 1995, he has coordinated the European ACTS project "VIDAS," concerned with the application of MPEG-4 technologies in multimedia telecommunication products. He is the author of more than 60 scientific papers in the area of multimedia data management and coding.

Roberto Pockaj was born in Genova, Italy, in 1967. He received the master's degree in electronic engineering from the University of Genova, Italy, in 1993, where he is currently pursuing the Ph.D. degree in the Department of Communications, Computer and Systems Science (DIST).

From June 1992 to June 1996 he was with the Marconi Software Co. as a Software Designer in the research/development area of real-time image and signal processing for optoelectric applications (active and passive laser sensors). Since September 1996, he has been involved in the definition of the new standard MPEG-4 for the coding of multimedia contents within the ad hoc groups on synthetic and natural hybrid coding (SNHC) and face and body animation (FBA).