MPEG-4 Facial Animation Technology: Survey, Implementation, and Results

Gabriel Antunes Abrantes, *Student Member, IEEE*, and Fernando Pereira, *Member, IEEE*

***Abstract*—The emerging MPEG-4 standard specifies an object-based audiovisual representation framework, integrating both natural and synthetic content. Tools supporting three-dimensional facial animation will be standardized for the first time. To support facial animation decoders with different degrees of complexity, MPEG-4 uses a profiling strategy, which foresees the specification of object types, profiles, and levels adequate to the various relevant application classes. This paper first gives an overview of the MPEG-4 facial animation technology. Subsequently, the paper describes the Instituto Superior Técnico implementation of an MPEG-4 facial animation system, then briefly evaluates the performance of the various tools standardized, using the MPEG-4 test material.**

***Index Terms*—Audiovisual objects, facial animation, MPEG-4.**

I. Introduction

THE FAST evolution of digital technology in the last decade has deeply transformed the way by which information, notably visual information, is generated, processed, transmitted, stored, and consumed. Nowadays, more and more applications include visual information, and the user is becoming more and more interactive in his relationship with visual information. With the digital revolution, it became possible to further exploit a well-known concept: the more that is known about the content, the better can be its representation, processing, etc., in terms of efficiency, efficacy, and allowed functionalities. In fact, in the world of natural visual data (video), strong limitations result from the way by which video data are acquired and subsequently represented, the so-called frame-based representation. This video data model is the basis of all the analog and digital video representation standards available today, namely, PAL, SECAM, NTSC, H.261, H263, MPEG-1, and MPEG-2.

Recognizing that audiovisual content should be represented using a framework that is able to give the user as many real-world-like capabilities as possible, the Moving Pictures Experts Group (MPEG) decided in 1993 to launch a new project, known as MPEG-4. MPEG-4 is the first audiovisual representation standard modeling an audiovisual scene as a composition of audiovisual objects with specific characteristics and behavior, notably, in space and time [1], [2]. The object composition approach allows MPEG-4 to support new functionalities, such as content-based interaction and manipulation, as well as improvements to already available functionalities, such as coding efficiency and error resilience, by using for each type of data the most adequate coding technology [2].

One of the most exciting and powerful consequences of the object-based approach is the integration of natural and synthetic content. Until now, the natural and synthetic audiovisual worlds have evolved quite in parallel. The MPEG-4 representation approach allows the composition of natural and synthetic data in the same scene, unifying the two separate, but complementary, worlds. This unification allows MPEG-4 to efficiently represent natural as well as synthetic visual data, without undue translations like the conversion to pixel-based representations of synthetic models. Another powerful consequence of this strategy is that the conversion to the pixel and audio sample domains of a composition of various natural and synthetic objects is deferred to the receiving terminal, where locally specified user controls and viewing/listening conditions may determine the final content.

To fulfill the objectives proposed, notably in the area of synthetic content, MPEG created a new subgroup called Synthetic and Natural Hybrid Coding (SNHC), which had the task of addressing the issues related to synthetic data, notably, representation and synchronization. After a long collaborative process, MPEG-4 Version 1 reached, in October 1998, final draft international standard (FDIS) status, which is the very last stage of an ISO international standard (IS), including technology that is fully mature and deeply tested. The SNHC topics that found their way into the MPEG-4 systems [3], visual [4], and audio [5] final draft international standards are three-dimensional (3-D) facial animation, wavelet texture coding, mesh coding with texture mapping, media integration of text and graphics, text-to-speech synthesis (TTS), and structured audio. The SNHC technology standardized in MPEG-4 Version 1 will support applications such as multimedia broadcasting and presentations, virtual talking humans, advanced interpersonal communication systems, games, storytelling, language teaching, speech rehabilitation, teleshopping, telelearning, etc., based on or including text and two-dimensional (2-D)/3-D graphics capabilities. More ambitious goals will be pursued with MPEG-4 Version 2 [6], which will complement Version

Manuscript received October 1, 1997; revised November 1, 1998. This work was supported by the European Commission under ACTS project MoMuSys. The work of G. A. Abrantes was supported by the Junta Nacional de Investigação Científica e Tecnológica, Portugal. 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 Instituto Superior Técnico, Instituto de Telecomunicações, Lisboa 1049-001 Portugal (e-mail: Gabriel.Abrantes@lx.it.pt; Fernando.Pereira@lx.it.pt).

Publisher Item Identifier S 1051-8215(99)02332-0.

1 by including new tools, providing additional functionalities such as body animation. Each stage of MPEG-4 Version 2 is foreseen to happen about one year after the corresponding stage for Version 1.

Among the technologies to be standardized in MPEG-4 Version 1 and developed in the SNHC subgroup, 3-D facial animation assumes a special relevance since the use of 3-D model-based coding applied to human facial models may bring significant advantages, notably for critical bandwidth conditions. The standardization of the parts of this technology, essential to guarantee interoperability, may significantly accelerate the deployment of applications using synthetic human heads, which represent real or fictitious humans. The animation of 3-D facial models requires animation data, which may be synthetically generated or extracted by analysis from real faces, depending on the application. Analysis is usually a complex task with precise constraints, which strongly depend on the application conditions. As a consequence, analysis may be real-time or off-line, fully automatic or human guided. While videotelephony-like applications typically require real-time and fully automatic facial analysis, storytelling applications may allow off-line, human-guided analysis. Other applications, such as teleshopping and gaming, may not even need any analysis at all since there may be no intention to reproduce a real face but just to create an entertaining face, which may be accomplished by means of a facial animation editor.

Besides giving an overview of the MPEG-4 facial animation technology, this paper describes the implementation of the MPEG-4 facial animation system developed at the Instituto Superior Técnico (IST), Universidade Técnica de Lisboa. The system follows the MPEG-4 visual and systems FDIS, issued in October 1998 [3], [4]. In Section II, the facial animation technology included in the MPEG-4 Version 1 FDIS, as well as the decisions regarding facial animation profiling, will be described. While Section III will describe the MPEG-4 facial animation system implemented at IST, Section IV will present results, allowing a first evaluation of MPEG-4 facial animation technology.

II. Basics on the MPEG-4 Architecture

To reach the objectives described above, an MPEG-4 system needs to go beyond the capabilities of the more traditional audiovisual representation standards, where the audiovisual scenes were always composed of a frame-based video sequence and the corresponding audio data. MPEG-4 integrates visual objects of various types, e.g., frames, 2-D arbitrarily shaped video objects, text, graphics, and 3-D faces, as well as audio objects of various types, e.g., music, speech, and structured audio. Consequently, there is the need to consider not only the coded representations of the various audiovisual objects but also some information describing the way by which the various objects are composed to build the final audiovisual scene. To fulfill this need, MPEG-4 organizes the data in terms of elementary streams, which may contain different types of information [3]:

1) audiovisual coded data such as 2-D arbitrarily shaped video, music, and speech associated with natural video and audio objects;
2) facial animation data associated with 3-D facial objects;
3) scene description information addressing the spatio-temporal positioning of the media objects in a scene as well as user interaction;
4) object content information (OCI) providing textual descriptive information about the events associated with the scene and the individual objects;
5) object descriptors, mainly describing the type of content in each individual elementary stream.

While at the sending side the several elementary streams associated with the various objects are multiplexed together, at the receiving side the elementary streams are demultiplexed, the various media objects are decoded, and the scene is composed using the scene-description information. This scene-description information is at the heart of the MPEG-4 vision and thus at the core of most of the new functionalities that MPEG-4 can provide. In MPEG-4, the description of the scene follows a hierarchical structure, which can be represented by a graph, the scene graph. Each node of the graph is a scene object, and thus a 3-D face object will also be a node in the scene graph. The graph structure is not static, which means that relationships can dynamically change and nodes can be added or deleted. The objects can be located in a 2-D or 3-D space.

The architecture described above requires MPEG-4 to address not only the coding of the raw audiovisual data, and the facial animation data, but also the coding of the scene description information. The scene description coding format, specified by MPEG-4, is known as the binary format for scene description (BIFS) and represents a predefined set of scene object types, e.g., video, audio, 3-D faces, and corresponding behaviors along with their spatio-temporal relationships [3]. BIFS consists of a collection of nodes, which describes the scene and its layout. An object in the scene is described by one or more BIFS nodes, which may be grouped together using a grouping node. A scene object that has an associated media stream is called a media object, while a video object is a media object because it has an associated stream with its coded texture. A text object is not a media object since all its characteristics and behavior are defined at the BIFS level. As will be seen in the following, a 3-D face object is a media object since it will have an associated media stream, the facial animation stream.

III. The MPEG-4 Facial Animation Tools

Taking into account the relevance of the applications and the maturity of facial animation technology, MPEG-4 decided to standardize the necessary tools to allow the provision of a new range of applications relying on standardized facial animation technology. The face object specified by MPEG-4 is a representation of the human face structured in a way that the visual manifestations of speech are intelligible, the facial expressions allow recognition of the speaker's mood, and reproduction of a real speaker as faithfully as possible is supported [4]. To fulfill these objectives, MPEG-4 specified three types of facial data.

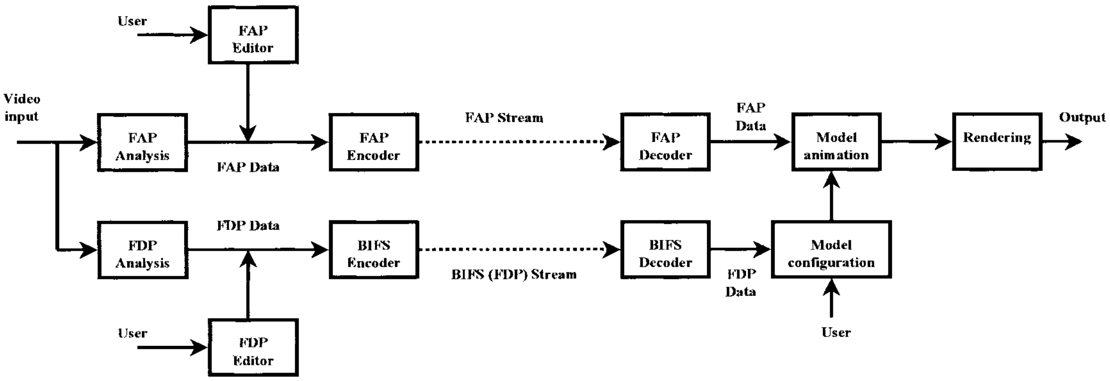


Fig. 1. General block diagram of a 3-D facial animation system.

1) *Facial Animation Parameters (FAP's):* FAP's allow one to animate a 3-D facial model available at the receiver. The way by which this model is made available at the receiver is not relevant. FAP's allow the animation of key feature points in the model, independently or in groups, as well as the reproduction of visemes and expressions.
2) *Facial Definition Parameters (FDP's):* FDP's allow one to configure the 3-D facial model to be used at the receiver, either by adapting a previously available model or by sending a new model. The new or the adapted model is then animated by means of FAP's.
3) *FAP Interpolation Table (FIT):* FIT allows one to define the interpolation rules for the FAP's that have to be interpolated at the decoder. The 3-D model is then animated using the FAP's sent and the FAP's interpolated according to the FIT.

While FAP's continuously provide visual information associated with the behavior of the 3-D model, FDP's provide model configuration information, which is typically sent only once. For this reason, FAP's are coded as an individual elementary stream—*the facial animation stream*—while FDP's are fully coded as BIFS nodes and thus are sent in the BIFS stream.

Since the strength of a standard is associated with the degree of interoperability it provides, but also with its flexibility and its ability to cope with further technological developments, it is essential that a standard only specifies the tools for which standardization is essential for interworking. This strategy allows interoperability while maximizing the evolution and competition opportunities. As happened in the past for other standards, this approach has important consequences in terms of the tools that are to be standardized in the facial animation area. The nonstandardization of tools nonessential to guaranteeing interoperability allows the standard to continuously integrate the relevant technological developments in all the nonnormative areas, as well as to provide opportunities for the industry to compete, which is always important to stimulate the development of better products.

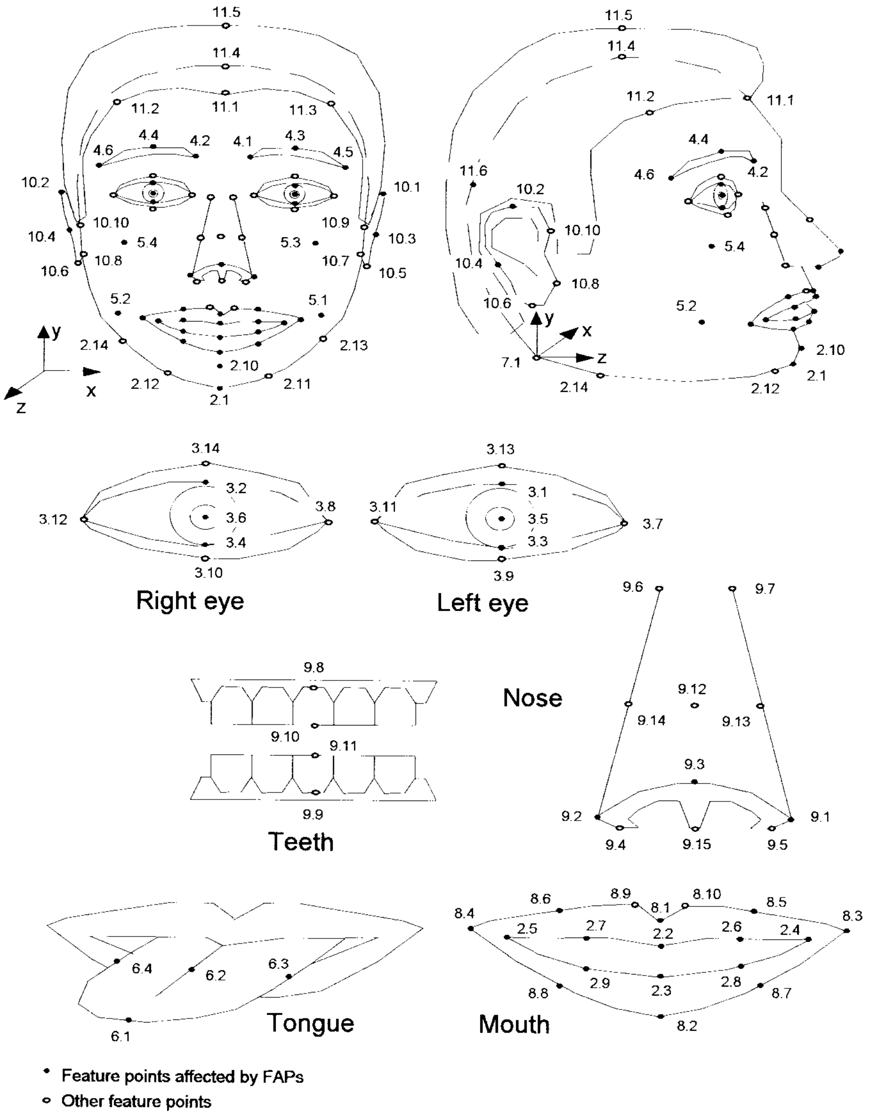
Fig. 1 shows a general block diagram of a 3-D facial animation system, which may fit many applications. While for some applications FAP's and FDP's will be extracted from a real video input, for many others the analysis phase does not exist. This means that FAP and FDP data are artificially edited to fulfill a certain goal by synthesizing the necessary data. The way by which FAP's and FDP's are generated and coded—automatically or manually, real-time or nonreal-time—is completely irrelevant to the receiving terminal, provided that the FAP's and FDP's follow the standardized bitstream syntax and semantics.

MPEG-4 decided not to standardize the 3-D facial model, under the assumption that FAP's can provide good animation results with any reasonable model; however, whenever needed, face models can be configured using FDP's. This decision allowed MPEG-4 to avoid the difficult normative choice of a unique 3-D facial model, providing a flexible solution in terms of the type and complexity of the facial models that can be used without interoperability problems. Due to this freedom in terms of the 3-D facial model, it is difficult for the sender to know precisely the appearance of the synthesized result at the receiver since a large range of models may be used. This fact led to the specification of a minimum set of decoding, animation, and adaptation rules, under the assumption that the receiver will always do the best it can with the received information for which few strict rules are defined. Finally, and as it happens for any MPEG-4 visual object, the rendering is not standardized, depending on the rendering algorithms available at the receiver as well as on its computational power. Although for MPEG-4 Version 1 the composition of objects is nonnormative, this approach may have to be changed in the future, at least for the solutions addressing the applications where the sender wants to know precisely what the final result at the receiver will be, such as some broadcasting applications.

To avoid burdening the MPEG-4 terminals willing to support 3-D face objects with all the facial animation tools, the MPEG-4 profiling mechanisms will be used. This will allow the deployment of more or less powerful and complex facial animation enabled terminals, depending on the number of facial animation tools that they are able to understand and on the complexity of the bitstreams that they are able to handle.

A. Facial Animation Parameters

This section and the corresponding one for FDP's intend to address the most relevant normative elements associated with the FAP syntax, semantics, and decoding. As mentioned



Right eye

Left eye

Nose

Teeth

Tongue

Mouth

Fig. 2. Feature-points grouping [4].

before, FAP's were designed to allow the animation of faces, reproducing movements, expressions, emotions, and speech pronunciation. FAP's are based on the study of minimal facial actions and are closely related to muscle actions. The chosen set of FAP's represents a complete set of basic facial movements, allowing terminals to represent most natural facial actions as well as, by exaggeration, some nonhuman-like actions useful, e.g., for cartoon-like animations.

The FAP set includes 68 FAP's, 66 low-level parameters associated with the lips, jaw, eyes, mouth, cheek, nose, etc., and two high-level parameters (FAP's 1 and 2) associated with expressions and visemes [4]. While low-level FAP's are associated with movements of key facial zones, typically referenced by a feature point (see Fig. 2), as well as with rotation of the head and eyeballs, expressions and visemes represent more complex actions, typically associated with a set of FAP's. Low-level FAP's are grouped as defined in Fig. 2. Although the encoder knows the reference feature point for each low-level FAP, it does not precisely know how the decoder will move the model vertices around that feature point, this means the FAP interpretation model, that describes the specification of the precise changes in the model vertices corresponding to each FAP.

All low-level FAP's involving translational movement are expressed in terms of facial animation parameter units (FAPU's). These units are defined in order to allow the interpretation of the FAP's on any facial model (normally unknown to the sender) in a consistent way, producing reasonable animation results. FAPU's correspond to fractions of distances between some key features, e.g., mouth–nose separation, eye separation, etc. [4]. The fractional units used for the various FAP's are chosen to allow enough precision for the corresponding FAP.

Table I shows an excerpt from the FAP list, including number and name, a short description, the specification of the associated FAPU (e.g., MNS is mouth–nose separation and MW is mouth width), whether the FAP is unidirectional or bidirectional, the definition of the movement direction for positive values, the group number, the FDP subgroup number, and the default quantization step size.

TABLE I
Excerpt of the FAP Specification Table [4]

#	FAP name	FAP description	Units	Uni- or Bidir	Motion	Group	FDP sub-group number	Default quantiz. Step
1	Viseme	Set of values determining the mixture of two visemes (e.g. pbm, fv, th)	na	na	na	1	Na	1
2	Expression	Set of values determining the mixture of two facial expressions	na	na	na	1	Na	1
3	Open_jaw	Vertical jaw displacement (does not affect mouth opening)	MNS	U	down	2	1	4
4	Lower_t_midlip	Vertical top middle inner lip displacement	MNS	B	down	2	2	2
5	Raise_b_midlip	Vertical bottom middle inner lip displacement	MNS	B	up	2	3	2

With the expression FAP, it is possible to select among six different expressions, namely, joy, sadness, anger, fear, disgust, and surprise. Visemes are the visual analog to phonemes and allow the efficient rendering of visemes for better speech pronunciation, as an alternative to having them represented using a set of low-level FAP's.

Zero-valued FAP's correspond to a neutral face; the receiving model is supposed to be in the neutral position at the beginning of a session. All FAP's are expressed as displacements from the positions defined for the neutral face, and thus it is essential to start from neutral faces that are as similar as possible. According to the specification [4], the neutral face is characterized by having all muscles relaxed, eyelids tangent to the iris, pupils as one-third of the iris diameter, lips in contact, mouth closed with the upper teeth touching the lower ones, and the tongue flat, horizontal, with its tip touching the boundary between upper and lower teeth.

The MPEG-4 specification allows one to indicate, in the FAP stream, the desired gender of the facial model to be used by the receiver. This information does not supersede the FDP information (if available) and is only provided as a "wish," thus, without any normative constraints for the decoder.

Following an important principle in MPEG standards, the MPEG-4 standard does not specify the FAP encoding process but only the FAP bitstream syntax and semantics, together with the corresponding decoding rules. The motivation for the use of FAP compression algorithms is to reduce the bit rate necessary to represent a certain amount of animation data with a certain predefined quality or to achieve the best quality for that data with the available amount of resources (bit rate). FAP's are coded at a certain frame rate, indicated to the receiver, which can be changed during the session. Moreover, one or more time instants can be skipped when encoding at a certain frame rate. Since not all the FAP's are used all the time, a FAP masking scheme is used to select the relevant FAP's for each time instant. FAP masking is done using a two-level mask hierarchy. The first level indicates, for each FAP group (see Fig. 2), one of the following four options (two bits).

1) No FAP's are coded for the corresponding group.
2) A mask is given indicating which FAP's in the corresponding group are coded. FAP's not selected by the group mask retain their previous value, if any value has been previously set (no interpolation is allowed).
3) A mask is given indicating which FAP's in the corresponding group are coded. The decoder should interpolate FAP's not selected by the group mask.
4) All FAP's in the group are coded.

For each group, the second-level mask indicates which specific FAP's are represented in the bitstream, where a "1" indicates that the corresponding FAP is present in the bitstream.

For the high-level FAP's (visemes and expressions), not only a FAP intensity value is sent, as for the rest of the FAP's. The viseme/expression FAP's allow two visemes/expressions from a standard set to be mixed together, and thus what is sent are the viseme/expression selection values and the corresponding intensities. To avoid ambiguities, the viseme/expression FAP's can only have an impact on low-level FAP's that are allowed to be interpolated at the time the viseme/expression FAP is applied.

FAP's can be coded as a sequence of face object planes, each corresponding to a certain time instant, or as a sequence of face object plane groups, each composed of a sequence of 16 face object planes, also called segments. Depending on the chosen mode, face object planes or face object plane groups, FAP's will be coded using a frame-based coding mode or a DCT-based coding mode. Compared to the frame-based mode, the DCT-based mode can give, in some conditions, higher coding efficiency at the cost of higher delay.

1) Frame-Based Coding Mode: In this mode, FAP's are differentially encoded and quantized using an adequate quantization step. Whenever desired, FAP's can be coded without prediction—intracoding mode. The default quantization steps of all FAP's may be scaled by means of a quantization scaling factor, ranging from one to 31. If lossless coding is to be used, the quantization scaling factor should be made equal to zero. The resulting symbols are then arithmetically encoded. At the decoder, FAP values are set according to one of three cases: by a value in the bitstream, by a retained value previously set by the bitstream, or by means of decoder interpolation. FAP values set by the bitstream and subsequently masked are allowed to be interpolated only if an explicit indication in this sense is sent. For the high-level FAP's, the intensities are encoded as for the other FAP's, but the visemes/expression selection values are differentially encoded without arithmetic encoding (and of course quantization).

2) DCT-Based Coding Mode: In this mode, 16 face object planes are buffered and DCT encoded, using intra- or intercoding. The DC coefficient is then differentially coded and quantized using a quantization step, which is one-third of the quantization step for the AC coefficients. It is possible to adjust the quantization steps to be used by means of a scaling index, varying from zero to 31, indexing a table with scaling factors for the default quantization step values. The coefficients are then coded using Huffman tables. High-level FAP intensities and selection values are only differentially encoded without entropy coding.

Independently of the FAP coding mode used, all MPEG-4 facial animation decoders are required to generate at their output a facial model including all the feature points defined in this specification, even if some of the features points will not be affected by any information received from the encoder.

B. Facial Definition Parameters

FDP's were designed to allow the customization of a proprietary 3-D facial model resident at the receiver or to download a completely new model together with the information about how to animate it. While the first option allows limited control by the sender on what the animated face will look like, the second option should allow full control of the animation results. Since FDP's have the purpose of configuring the facial model, they are typically sent only once per session, although for some applications, FDP's may be used along the session to achieve some particular effects, like face deformation, texture changing, etc. As said before, FDP's are encoded in BIFS, using the nodes specified in the MPEG-4 systems standard [3]. To avoid ambiguous situations and unexpected results due to the (many times unknown) interpolation rules used by the decoder, the sender should send a convenient amount of FDP configuration data. Depending on the FDP data used, four relevant cases in terms of the implementation of an MPEG-4 facial animation system can be identified.

1) No FDP Data (Only FAP Data): Since no FDP data are sent, the proprietary 3-D model residing at the receiver is animated with the received FAP data without performing any model adaptation.

2) Feature Points: A set of feature points as defined in Fig. 2, and represented by means of their 3-D coordinates, is sent to the receiver with the purpose of calibrating the resident model. These feature points have to correspond to a neutral face, as previously described. Due to the possible difficulties in obtaining some feature points, it is not required to send all feature points, and thus subsets are allowed. The receiver has to adapt the model in a way that the vertices on its proprietary model corresponding to feature points must coincide with the feature points received. No further rules to constrain this calibration process are specified.

3) Feature Points and Texture: It is well known that texture mapping may significantly improve the realistic appearance of an animated model. This case considers sending 3-D feature points to adapt the model geometry (as explained above) as well as texture together with corresponding calibration information (2-D feature points) to allow the decoder to further adapt the model to the texture to be mapped. For this purpose, one texture and 2-D texture coordinates corresponding to some feature points are sent. To improve texture mapping, the sender can indicate which type of texture is being sent: either a cylindrical projection (e.g., cyberware texture) or an orthographic projection (e.g., frontal texture). No further rules to constrain this calibration and mapping process are specified.

4) Facial Animation Tables (FAT's) and New 3-D Model: This is theoretically the only case where the sender can have full control over the animation results. To this end, a 3-D facial model is downloaded to the receiver, together with feature points (to allow the extraction of FAPU's) and FAT's which define the FAP behavior for the new model. The FAP behavior in the newly received model is specified by indicating which and how the new model vertices should be moved for each FAP. The downloaded model can be composed of multiple meshes, each with an associated texture.

C. FAP Interpolation Table

The FIT allows a smaller set of FAP's to be sent during a facial animation session since it provides the sender a tool to specify interpolation rules for some or all the FAP's that are set for interpolation by the receiver (and thus not received). The small set of FAP's transmitted can then be used to determine the values of other FAP's, using a rational polynomial mapping between parameters. For example, the top-inner-lip FAP's can be sent and then used to determine the top-outer-lip FAP's. To this end, the sender specifies a FAP interpolation graph and a set of rational polynomial functions that specify which and how sets of received FAP's are used to determine other sets of FAP's. Having more FAP's defined following encoder indications (in a cheap way) allows the sender to have a higher degree of control over the animation results. For more details, see the MPEG-4 system's FDIS [3].

D. BIFS for Facial Animation

In MPEG-4, the scene description information is represented using a parametric methodology, BIFS [3]. The description

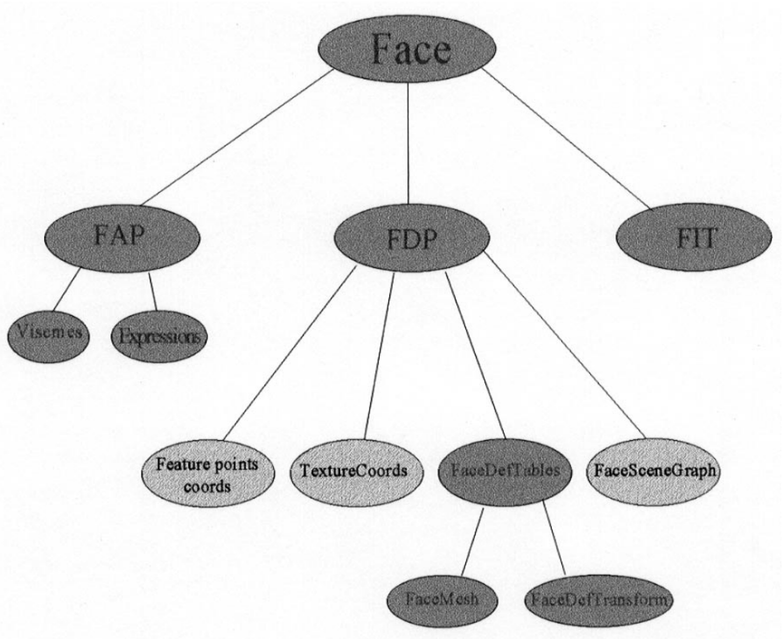


Fig. 3. Facial animation BIFS nodes.

consists of an efficiently encoded hierarchy (tree) of nodes with attributes and other information, including event sources and targets. Leaf nodes in the tree correspond to particular audio or visual objects (media nodes), whereas intermediate nodes perform grouping, transformation, and other operations (scene description nodes). The scene description can evolve over time by using description updates. The MPEG-4 scene description framework is partly based on the virtual reality modeling language, which has been significantly extended, e.g., to address streaming and synchronization issues. To offer complete support for face and body animation, BIFS defines a set of face and body nodes [3]. Although the standardization of body animation tools will only happen in MPEG-4 Version 2, BIFS already considers the necessary nodes to guarantee a good integration with the facial animation nodes. Fig. 3 shows the most important BIFS nodes for facial animation.

The face node organizes the animation and adaptation of a face. The FAP node shall be always specified since it contains the most essential facial animation data, the FAP's. The FDP node defines the particular shape and appearance of a face by means of adaptation data (featurePointsCoord, textureCoords, and the texture in faceSceneGraph) or an entire new model and FAP animation rules (faceSceneGraph and faceDefTables). If the FDP node is not specified, the default face model of the decoder is used. The FIT node, when specified, allows a set of unreceived FAP's to be defined in terms of a set of received FAP's. Last, renderedFace is the scene graph of the face after it is rendered (all FAP's and FDP's applied). The standard specifies processes that involve the reading of node values, e.g., FAP's, and then the writing of output values to nodes in the face hierarchy, e.g., renderedFace. For more details, see the MPEG-4 systems specification [3].

E. Object Types and Profiles Including Facial Animation

Since MPEG-4 will standardize a large number of tools that can be used requiring more or less computational power and memory at the receiver, it is essential to provide a mechanism that avoids an extreme situation where every MPEG-4 terminal would be obliged to decode all types of bitstreams, in terms of syntax and complexity. To allow the specification of less complex (and, in principle, less expensive) MPEG-4 terminals, targeted to a certain class of applications, while still guaranteeing interoperability with their peers, MPEG-4 defined a profiling strategy based on two major concepts: object types and profiles [4].

1) Concepts: The object type defines the syntax of the elementary stream for a single object corresponding to a meaningful entity in the scene. This basically means that an object type defines the list of tools that can be used for a certain type of object. Object types are defined for audio and visual objects.

As MPEG-4 wants to integrate, in the same terminal/scene, several types of objects that correspond to elementary streams with different syntactic characteristics, a higher layer above object type has to be defined, which is associated to the syntactic characterization of the terminal capabilities, the profiles. A profile defines the set of tools that can be used in a certain MPEG-4 terminal. Audio and visual profiles are defined in terms of the audio and visual object types that can be understood; they implicitly have associated certain BIFS (media) nodes. Graphics profiles define, in terms of BIFS nodes, which graphical elements can be used in the scene. Scene description profiles define the scene description capabilities allowed in terms of BIFS nodes (nonmedia nodes). Object descriptor profiles define the terminal capabilities in terms of the object descriptor and sync layer tools. The audio, visual, and graphics profiles can be generally classified as media profiles since they are associated with the media elements in the scene, while the scene description profiles are associated with the composition capabilities.

Since profiles only define the tools, and thus the syntax, and not complexity bounds, levels of complexity have to be defined for each profile. A level is a specification of the constraints (e.g., bit rate, sampling rate, memory size, etc.) on the audio, visual, graphics, scene description, and object descriptor profiles and thus on the corresponding tools. Levels are only defined for profiles and not for object types under the assumption that the terminals will share the available resources among the various audio or visual objects.

2) Profiling Situation for Facial Animation: According to the MPEG-4 visual FDIS [4], one facial animation object type has already been defined: the simple face object type. In terms of visual profiles, three profiles include the simple face object type: the simple facial animation profile, the basic animated 2-D texture, and the hybrid profile.

Since facial animation object types will be the elementary decoding entities to implement in terms of MPEG-4 facial animation enabled systems, the main characteristics of the already defined object type will be described in the following. The simple face object type is the only standardized facial animation object type in MPEG-4 Version 1, and also the simplest useful object type that can be designed with the tools specified. This facial animation object type is mainly characterized by the fact that the decoder is only obliged to use the FAP data received. This means that even if FDP data are received (the face node associated to this object type includes FAP and FDP data), the decoder may ignore it or part of it by just ignoring the FDP data in the BIFS stream or by parsing the BIFS stream in order to just use the FDP data that it is able to handle. If only one side of a left–right pair of a FAP is received, this type of facial animation decoder must use the received FAP value for both left and right FAP values. Last, this type of facial animation decoder is required to use the viseme and expression FAP's (1 and 2). With this object type, the sender has no guarantee what the animated face will look like since it does not know or control the 3-D model being used.

The facial animation object types are to be included in the visual profiles providing solutions for the classes of applications where facial animation capabilities are required. It is also possible that facial animation capabilities are added to visual profiles where these capabilities are less essential, if this addition does not burden too much the profile in question. As mentioned above, there are three visual profiles in the MPEG-4 visual FDIS that include the simple face object type: the simple facial animation profile, the basic animated 2-D texture, and the hybrid profile. The simple facial animation profile includes only simple face objects and addresses applications where facial animation is the only relevant visual capability (no natural video capabilities). In terms of tools, this profile is completely characterized by the specification of the simple face object type since only this object type is allowed. According to the MPEG-4 visual FDIS [4], two levels have been defined for this profile.

The basic animated 2-D texture profile puts together facial animation capabilities and 2-D mesh animation with texture mapping capabilities. The hybrid profile is the most powerful visual profile, including most natural and synthetic visual object types, addressing content-rich multimedia applications. In terms of facial animation, the levels for these two profiles are based on the levels defined for the simple facial animation profile.

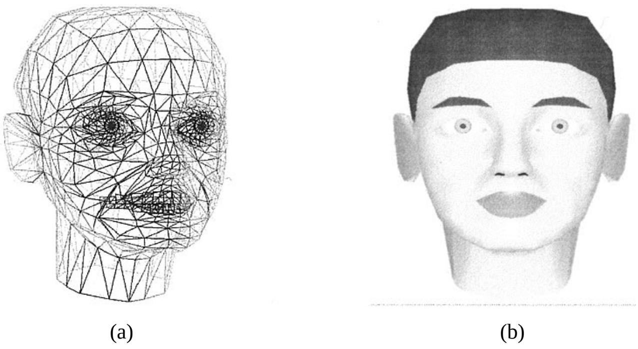

(a) (b)

Fig. 4. IST facial model (a) only with polygons and (b) with a simple texture.

IV. The IST MPEG-4 Facial Animation System

The MPEG-4 facial animation system developed at IST, following the specification in the MPEG-4 systems and visual FDIS's [3], [4], will be described in this section. The intention is to describe here the nonnormative parts of the system developed, complementing the normative parts already described in the previous section. The IST facial animation system is complete in terms of FAP streams, the encoder-decoder pair is working, but FDP data are still being obtained from FDP files and not from standard BIFS streams. An analysis module to extract FAP and FDP data from video has also been implemented but will not be addressed in this paper [7], [10].

A. The IST 3-D Facial Model

Since it is a basic assumption that all MPEG-4 facial animation decoders have their own 3-D facial model, any implementation needs to start by defining (or borrowing) a 3-D facial model. The IST 3-D facial model includes the complete human head and is a (very) modified version of a well-known model developed by Parke [8]. The Parke model was used just as a starting point since it only includes the frontal part of the head, missing other important parts, such as the back of the head, the tongue, and the ears. As these parts are important for some expressions and animations, it was decided to improve the Parke model, leading to an IST model including the back of the head, tongue, and ears, as well as other improvements, notably in the eyebrows and cheeks, in order to obtain more realistic animations. The IST model (see Fig. 4) is used in the IST MPEG-4 facial animation system described in this section. The model can be divided into various regions according to their texture. The most important regions are the hair, eyebrows, mouth, teeth, tongue, eyes, nose, and ears.

B. Implementation of the FAP Tool

FAP encoders and decoders for both coding modes, frame based and DCT based, were implemented, along with an

MPEG-4 facial animation system[1] able to animate and render the IST facial model, using the decoded FAP data. The encoder is able to encode all the FAP's and can control the coding frame rate by skipping the FAP's corresponding to the nonrelevant time instants. Moreover, the encoder may choose which coding mode to use, depending on the relevant application. The major nonnormative implementation issue regarding the FAP tool (to be used in all facial animation object types) is the FAP interpretation model, which means the specification of the precise changes in the model vertices corresponding to each FAP. The FAP interpretation model strongly determines the realism of the animation. Since this is a nonnormative element, it will become one of the main components distinguishing the performance of different facial animation enabled terminals. The FAP interpretation model developed for the IST facial animation system distinguishes the following.

1) FAP's Associated with Translation: For the FAP's involving translation, the vertices that have to be moved as well as the direction and magnitude of the movement are specified. The magnitude of the movements depends on the FAP received; the values to apply have been obtained by intensive animation testing of the model for each relevant FAP. Moreover the selection of vertices affected by each FAP has also been intensively tested, as well as the interaction between FAP's, especially when they affect the same vertices, e.g., the lip FAP's. The intensive study of the independent and joint impact of each FAP allowed the development of a FAP interpretation model with harmonious deformations and realistic animations.

2) FAP's Associated with Eyelids: The FAP's associated with eyelids (top and bottom) correspond to a special case of translation. While the y-coordinate is computed as indicated above for translations, the x- and z-coordinates are computed so as to guarantee that the eyelid vertices are over an imaginary sphere with the same center of the eyeball, although with a slightly larger radius.

3) FAP's Associated with Rotation: For each FAP involving rotation, the vertices that have to be rotated as well as the rotation axis are specified.

Regarding FAP interpolation, notably, for the FAP's not received but for which interpolation is allowed, simple left–right and inner–outer FAP interpolation schemes have been implemented (if only one member of a pair is received, its value is used for both), if these FAP's are set to be interpolated.

C. Implementation of the FDP Tools

Regarding the adaptation by the sender of the receiver's facial model, the IST facial animation system considers all the cases specified in the standard and described in Section III-B.

1) Model Adaptation with Feature Points: The 3-D model adaptation process using feature points is divided into two stages: 1) global adaptation of the model and 2) local adaptation of some key facial elements, notably, the eyes, ears, nose, and teeth, if the corresponding feature points are sent. The feature points have to correspond to a neutral face, since otherwise the adaptation results may be quite inadequate.

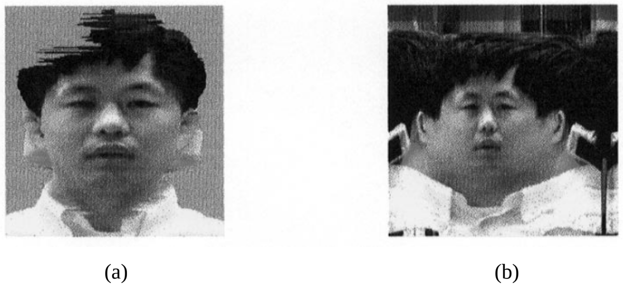


Fig. 5. *Chen:* (a) frontal texture and (b) cyberscan texture.

- *Global adaptation:* For most of the cases, the received set of feature points corresponds to a head that is very different from the receiver's model, in both size and 3-D position. To allow a good global reshaping of the 3-D model, it is essential to include a preprocessing step where resizing and 3-D repositioning of the model are performed before the global reshaping of the model. For further details, see [10].
- *Local adaptation:* The local adaptation stage has the target of better adapting the model in the regions corresponding to some key facial elements, notably, the eyeballs, eyelids, nose, ears, and teeth, using the adequate feature points, if received. For further details, see [10].

2) Model Adaptation with Feature Points and Texture: If a texture is sent together with the corresponding calibration information in the form of texture (2-D) feature points, for mapping on the receiver's model, a process similar to the one previously described for the model global adaptation is used. Notice that two sets of feature points may be received (or just one of them): one for model calibration (3-D coordinates) and another for texture mapping (2-D coordinates). The two sets do not have to include the same feature points since they are used for different purposes. As allowed by the MPEG-4 specification, the IST system considers two types of textures (see Fig. 5): 1) orthographic projection—frontal view of a face looking to a camera—and 2) cylindrical projection—the complete texture of the head (360° rotation) projected on a plan (cyberware).

Whenever the type of texture is specified, the most adequate mapping methodology is applied. Otherwise, the frontal-texture solution is applied as default since this is the most likely scenario. The major difference regarding the mapping of frontal and cyberscan textures is related to the projection of the 3-D model on the 2-D plane. For the rest, the adaptation process is the same for both cases. For further details, see [10].

3) Model Adaptation with FAT: The IST facial animation system may also change the FAP interpretation model to a new interpretation model described by means of FAT. The FAT specifies, for each selected FAP, the set of vertices to be affected in a new downloaded model, as well as the way they are affected. The downloaded FAP interpretation model

[1] The IST facial animation system was developed at IST in the context of the European project ACTS MoMuSys. Part of the software has been donated to ISO and subsequently included in MPEG-4 Part 5: Reference Software [9].

is then applied to the animation of the downloaded facial model, eventually after texture mapping. By sending a new 3-D facial model and the corresponding FAP interpretation model, the sender may completely control the result of the animation, notably, if all the FAP's used have been defined in the interpretation model. After the FAT receiving phase, all the animation works as already specified for the FAP tool.

V. Brief Evaluation of the MPEG-4 Facial Animation Tools

The study of the MPEG-4 facial animation specification would be incomplete without the analysis of some results for the tools standardized. This section intends to make a first evaluation of the MPEG-4 facial animation tools, using the IST implementation already described. To obtain the results presented in this section, the MPEG-4 facial animation test material has been used. For some cases, the test material does not fully comply with the ideal conditions—e.g., feature points corresponding to a nonneutral face—preventing the achievement of good results. The FAP files were encoded using the knowledge about the complete file (off-line processing); this means that FAP masks were set knowing in advance the FAP to be used, and the arithmetic encoder FAP ranges were also known.

A. Evaluation of the FAP Tool

This section evaluates the FAP tool in terms of the produced bit rates, FAP fidelity, and receiver's animation results, using the available FAP sequences. The FAP sequences have been encoded under various conditions in terms of frame rate, quantization steps, and coding modes. Also, the expression FAP is evaluated comparing some animation results with the original video material.

1) Bit Rate and FAP Peak Signal-to-Noise Ratio (PSNR) Performance: Since facial animation technology targets critical bandwidth conditions, it is essential to evaluate its performance in terms of bit rate and FAP fidelity. To plan for future applications, it is important to know the FAP fidelity versus bit rate tradeoff since this will affect the working conditions to be chosen. In this paper, the FAP fidelity is measured by means of the FAP PSNR, defined as

$$\mathrm{PSNR} = 10 \cdot \log \left(\frac{1}{N_{\mathrm{FAP}}} \sum_{i=1}^{N_{\mathrm{FAP}}} \left(R_i^2 / \mathrm{MSE}_i \right) \right)$$

$$\mathrm{MSE}_i = \frac{1}{N_{\mathrm{frame}}} \sum_{j=1}^{N_{\mathrm{frames}}} (fap\_orig[i][j] - fap\_recon[i][j])^2$$

where N_{FAP} is the number of FAP's used for a sequence, N_{frame} is the number of frames of the sequence, R_i the peak-to-peak range of the ith FAP, MSE_i is the mean square error along the sequence for each decoded FAP, and $fap\_orig[i][j]$ and $fap\_recon[i][j]$ are the original and decoded value of FAP i for frame j.

The bit rate versus quantization control parameter evolution for the frame-based and DCT-based modes is presented in Figs. 6 and 7 for the sequences *Expressions, Marco30,* and *Opossum2*. For the frame-based mode, the quantization scaling factor scales the default quantization step of each FAP by a factor ranging from one to 31, while for the DCT-based mode, the scaling index, varying from zero to 31, indexes a table with scaling factors for the default quantization step values. Figs. 6 and 7 show that the average bit rate can range from about 5 kbps to less than 0.5 kbps by just varying the quantization control parameter. The bit rate depends on the type of FAP content; the highly redundant, synthetic sequence *Expressions* achieves the lowest bit rates for the more relevant range of the quantization control parameters. The analysis of the animated sequences, using the IST facial animation system, indicates that for the higher values of the quantization control parameter, notably above 15 for the frame-based mode, the quality of the animation decreases significantly, especially for the eyebrows, eyelids, and mouth movements, becoming robot-like. This conclusion refers to the test sequences used and should not be simply extrapolated without more extensive testing. The results show that for both coding modes, the average bit rate does not decrease linearly with the quantization scaling factor, but there is rather a logarithmic evolution, and thus bit-rate savings become very low for higher values of the quantization control parameter. In conclusion, the use of very high values of the quantization control parameter does not bring significant advantages since the animation quality decreases without substantial bit-rate savings.

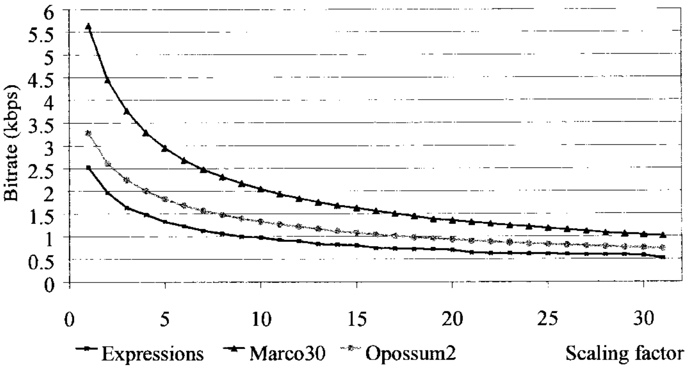


Fig. 6. Average bit rate versus quantization scaling factor for the sequences *Expressions, Marco30,* and *Opossum2* coded with the frame-based mode.

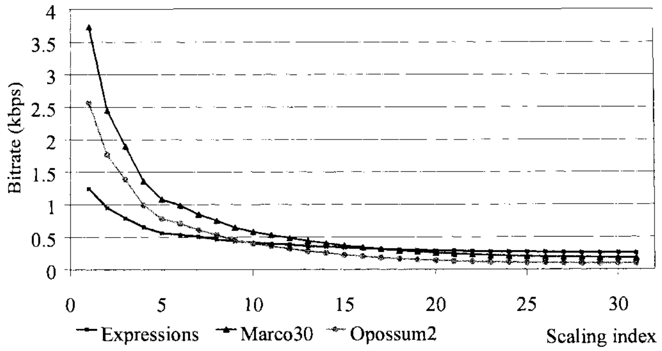


Fig. 7. Average bit rate versus quantization scaling index for the sequences *Expressions, Marco30,* and *Opossum2* coded with the DCT-based mode.

TABLE II
AVERAGE BIT RATE FOR VARIOUS CODING FRAME RATES USING THE TWO FAP CODING MODES

Sequence	Frame rate	Average bitrate (kbps)	
		Frame-based mode	DCT-based mode
Marco30	30 Hz	5.64	3.73
	15 Hz	3.66 (- 35%)	2.84 (- 24%)
	10 Hz	2.55 (- 55%)	2.36 (- 37%)
	7.5 Hz	2.07 (- 63%)	1.88 (- 50%)
Opossum2	30 Hz	3.28	2.57
	15 Hz	1.94 (- 41%)	1.85 (- 28%)
	10 Hz	1.39 (- 58%)	1.42 (- 45%)
	7.5 Hz	1.03 (- 69%)	1.13 (- 56%)
Marco30left	30 Hz	3.27	2.30
	15 Hz	2.19 (- 33%)	1.71 (- 26%)
	10 Hz	1.52 (- 53%)	1.46 (- 37%)
	7.5 Hz	1.25 (- 62%)	1.13 (- 51%)

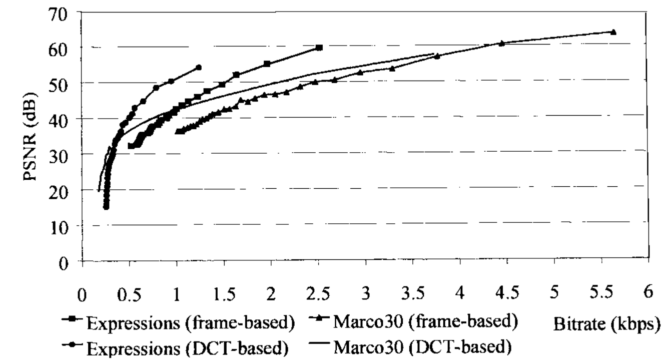


Fig. 8. PSNR versus bit rate for the sequences *Marco30* and *Expressions* coded at 25 Hz using the frame- and DCT-based coding modes.

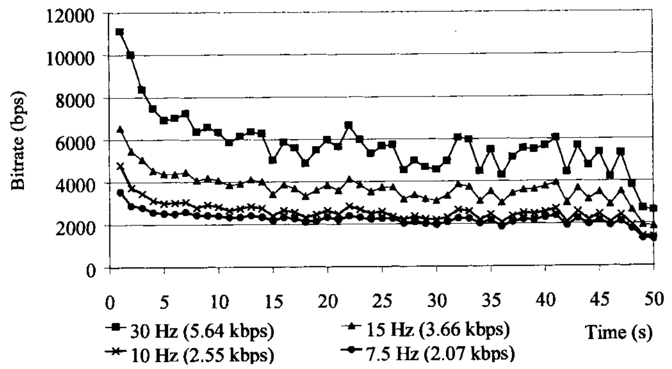


Fig. 9. *Marco 30:* bit-rate evolution for various frame rates using the frame-based coding mode.

For a more direct comparison of the two FAP coding modes, Fig. 8 shows the variation of the FAP PSNR versus the bit rate for the sequences *Marco30* and *Expressions* using the two coding modes. The results show that the DCT-based coding mode always outperforms the frame-based coding mode for the bit-rate range where both the solutions can work. However, the maximum PSNR achievable with the frame-based coding mode is higher than the maximum PSNR achievable with the DCT-based mode. The DCT-based coding mode PSNR gain depends on the bit-rate range and on the FAP content.

Since the price for the better coding efficiency performance of the DCT-based coding mode is initial delay (16 frames), only nonreal-time applications may typically benefit from it. For real-time applications, the frame-based coding mode is the only possible choice. In conclusion, the two FAP coding tools target different classes of applications, allowing the best possible performance to be achieved for both of them.

2) Influence of the Coding Frame Rate: The quantization step is not the only coding parameter supporting control of the bit rate produced. Another important parameter is the coding frame rate. In this section, the test sequences *Marco30* and *Opossum2* are coded at different frame rates to study the impact of the coding frame rate on the average bit rate using both coding modes. To allow a more complete comparison, the same sequence is also coded using the left–right symmetry tool, which means by just sending the left-side FAP when left and right FAP's are present (sequence *Marco30* left). Table II includes the average bit rates obtained for different frame rates using the two FAP coding modes (the quantization scaling factor and the scaling index were set to one). Figs. 9 and 10 show the evolution in time of the bit rate for *Marco30* for the two FAP coding modes using several coding frame rates.

The results obtained indicate that the average bit rate does not decrease linearly with the frame rate because although there are less FAP's to code, for the lower frame rates, the prediction error becomes higher, and thus more bits are spent.

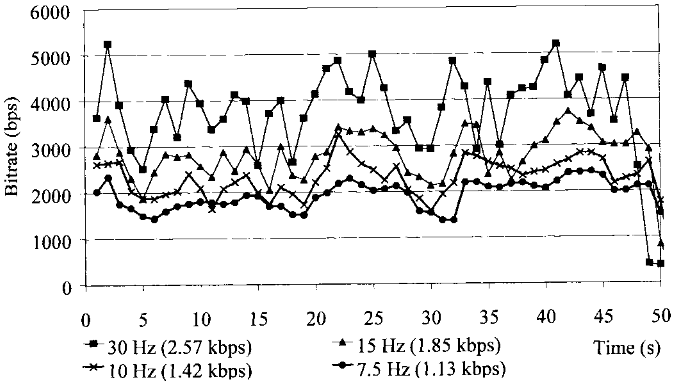


Fig. 10. *Marco 30:* bit-rate evolution for various frame rates using the DCT-based coding mode.

These two effects lead to an evolution where bit-rate savings decrease for lower frame rates. For the DCT coding mode, the decrease is even less significant since this mode takes higher benefit of the temporal correlation by using segments of 16 FAP face object planes; and for very low frame rates this correlation is low. The results also show the performance benefit of the left–right symmetry tool, which although very simple provides significant bit-rate gains. For example, at 30 Hz, there is a 40% decrease in terms of bit rate for *Marco30* for both coding modes with the quantization factors used. The results obtained show that this could be an efficient way to save bits and thus to make some difference between "compliant encoders." Although decreasing the coding frame rate decreases the bit rate more or less significantly, there is also the negative effect of low temporal resolution animations, which can only be fully evaluated by looking to the animated images. The choice of the most adequate frame rate should thus result from the tradeoff between bit-rate gains and animation jerkiness.

3) Evaluation of the Expressions: This section intends to show how the expression FAP can be used and what type of results can be obtained. If the encoder uses FAP 2, the expression FAP, just to select one expression (from those available) with a certain intensity, the decoder will use its own implementation (and understanding) of the relevant expression.

Fig. 11 shows the appearance corresponding to the six expressions defined by MPEG-4, according to the IST facial animation system (maximum intensity). It is interesting to note that this capability may automatically adjust the expressions according to the cultural environment of the receiver. This means that if an American sends a joy FAP command to Japan, the model will express joy in a Japanese way, if initially set to work in a Japanese environment, which means if the local expressions were set according to the Japanese culture. However, the encoder may also indicate to the receiver his understanding of each expression by indicating the set of low-level FAP's that builds each expression. To exemplify this, three FAP sets from three frames of the sequence *Emotions* corresponding to various expressions—notably, joy, anger, and surprise—are used to set the expressions at the receiver, with the results shown in Fig. 12.

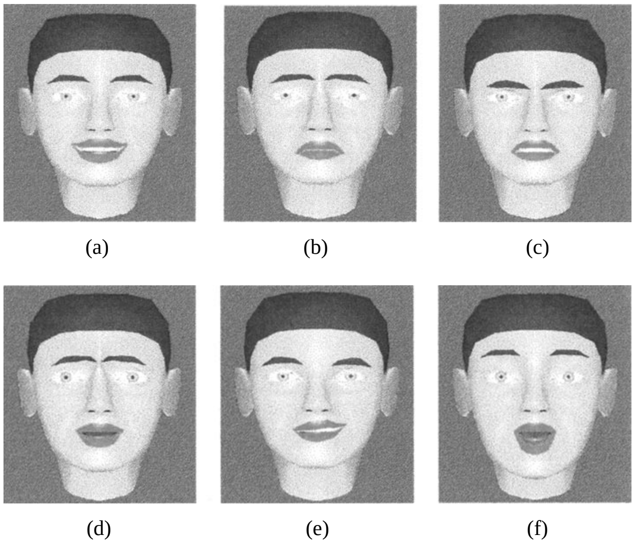


Fig. 11. IST implementation of the various expressions defined in FAP 2, at its maximum intensity: (a) joy, (b) sadness, (c) anger, (d) fear, (e) disgust, and (f) surprise.

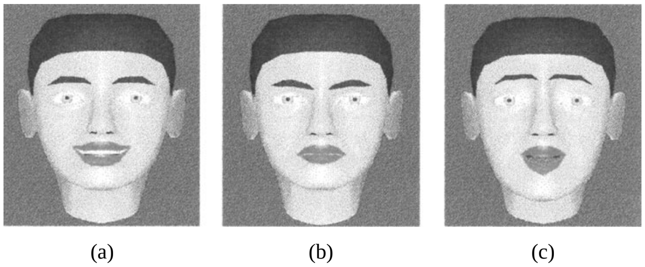


Fig. 12. *Expression* results using FAP 2 as set by the file *Emotions:* (a) joy, (b) anger, and (c) surprise.

The comparison between Figs. 11 and 12 gives an idea about how different the appearance for the various expressions may be, depending on their interpretation by different implementers. For example, the comparison of the expression *anger* implemented by the IST facial animation system [Fig. 11(c)] with the same expression resulting from the file *Emotions* [Fig. 12(b)] shows that the eyebrows are closer and the mouth is opened in the IST implementation. Another problem that may occur, and is visible in the expression "surprise" resulting from the sequence *Emotions* [Fig. 12(c)], is the production of strange shapes (see the mouth) due to the lack of knowledge by the sender of the receiver's model geometry as well as the limited knowledge of the FAP interpretation model. Moreover, since all FAP's are expressed in terms of FAP units, everything is dependent on these relative measurements and on all the ambiguities involved (sender and receiver may have slightly different measurement conditions), preventing a fully faithful reproduction of the intended facial actions. The use of the expression FAP may prevent most of these problems.

4) Animation with the IST Facial Animation System: Although it is (still) impossible to show in a paper what the animated sequences look like, it is at least possible to make a frame-by-frame comparison between the original sequence (if

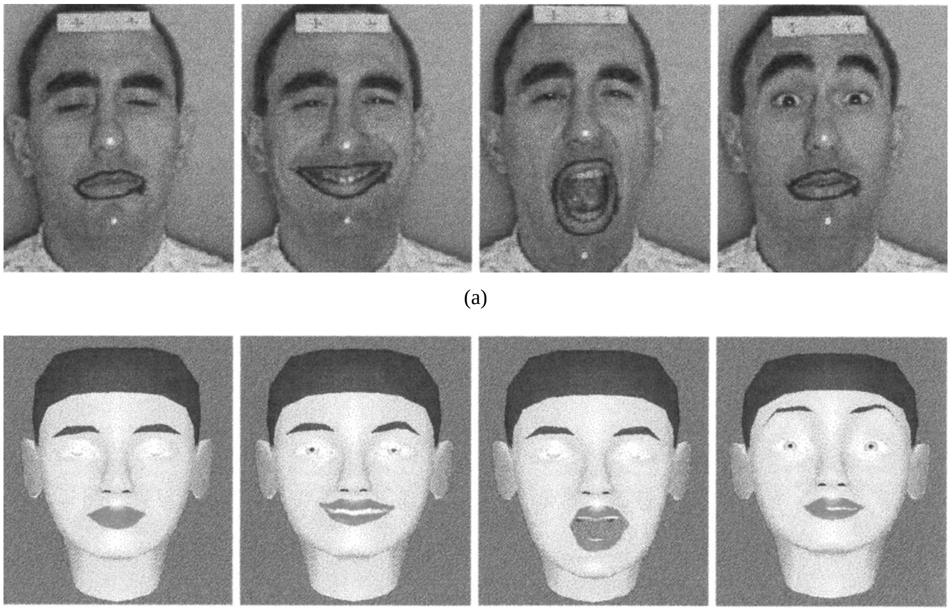


Fig. 13. Comparison of (a) original images for sequence *Marco30* and (b) corresponding animated frames using the IST animation system with the FAP sequence *Marco30.*

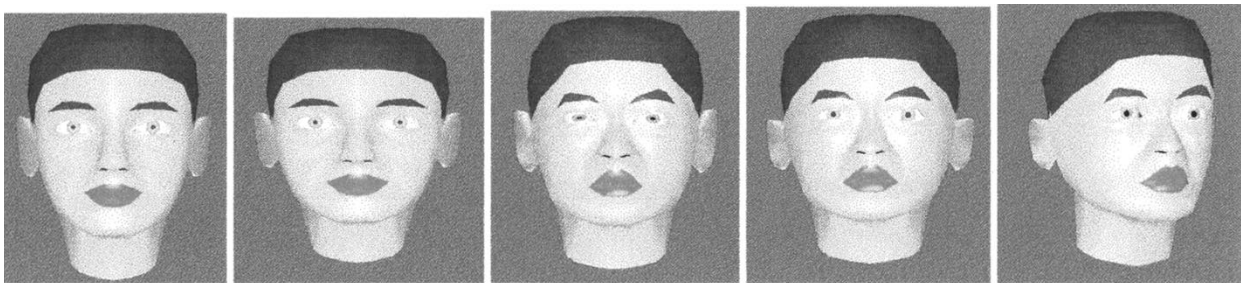


Fig. 14. Adaptation of the IST model with the feature points for *Chen:* (a) IST model, (b) resized model, and (c) global adaptation. Also shown are global and local adaptation from the (d) front view and (e) side view.

it exists) and the animated sequence. Fig. 13(a) shows a set of original frames from which the FAP file *Marco30*[2] has been extracted, while Fig. 13(b) shows the corresponding frames resulting from the animation with the IST facial animation system. These results show how similar to the real sequence the animated sequence can be when the simple facial animation profile is used. The absence of texture in the animated images influences the evaluation, but this is what will very likely happen with the simple face object type unless the receiver animates a model mapped with a locally stored texture. Also, the problems related to the FAP units' ambiguities may limit the faithfulness of the animation. In conclusion, it is quite difficult for FAP's to perfectly match the reality, and thus they only achieve an approximate result in comparison with the real sequence. However, this may be more than sufficient for many interesting applications.

B. Evaluation of the FDP Tools

This section evaluates the FDP tools in terms of the adaptation of the receiver model geometry and of the realism provided by texture mapping, notably for some expressions.

1) Model Adaptation with Feature Points: The results of applying the adaptation technique mentioned in Section IV-C1 to configure the geometry of the model residing at the receiver by using feature points are presented in this section. For the *Charles* and *Chen* cases, pictures for the various stages involved in the adaptation—resizing of model, global adaptation, and local adaptation—are shown (see Fig. 14). Since the feature points for *Chen* do not correspond to the specified neutral position (the head is rotated and the mouth is opened), they had to be processed to correspond to a neutral face. The results would very likely be better if the feature points initially corresponded to a neutral face.

Due to the difficulty of extracting good feature points and all the ambiguities/freedom involved in the adaptation process, it can be concluded that model adaptation with feature points provides only limited capabilities in terms of the control given

[2] The fact that the face in *Marco30* has been prepared to facilitate automatic FAP extraction does not imply that no good facial analysis exists that does not need such a preparation.

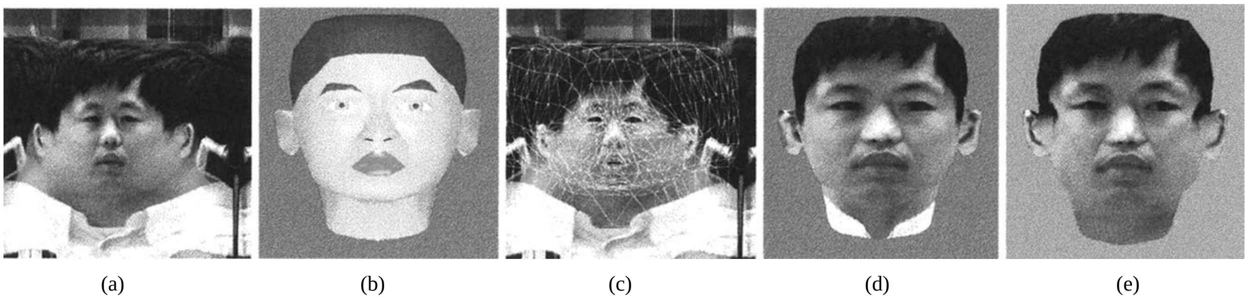


Fig. 15. Texture mapping for *Chen:* (a) cyberware texture, (b) adapted model without texture, (c) adapted model projection over the texture, (d) adapted model with texture, and (e) adapted model with texture if the received cyberware texture is taken as a frontal texture.

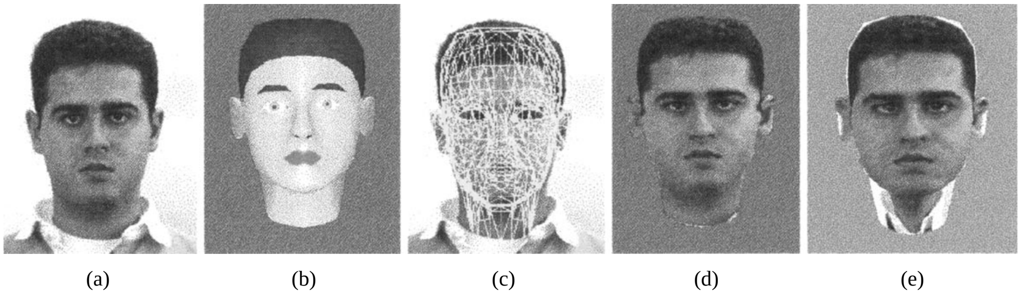


Fig. 16. Texture mapping for *Luis_IST:* (a) frontal texture, (b) adapted model without texture, (c) adapted model projection over the texture, (d) adapted model with texture, and (e) adapted model with texture if the received frontal texture is taken as a cyberware texture.

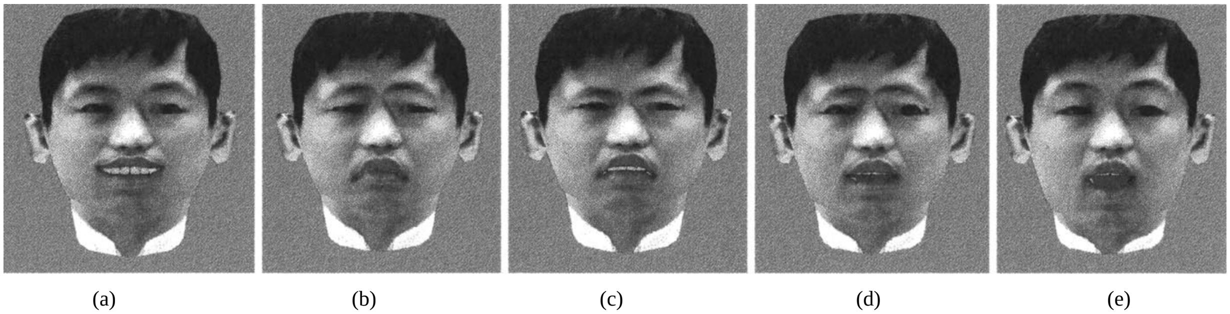


Fig. 17. *Chen* animated with the IST implementation of the expressions: (a) joy, (b) sad, (c) anger, (d) fear, and (e) surprise.

to the sender regarding the receiver's model geometry. These capabilities, however, may be sufficient for some relevant applications.

2) Model Adaptation with Feature Points and Texture: This section presents some results of adapting the geometry of a model with 3-D feature points and subsequently mapping a texture using a set of texture feature points, following the process mentioned in Section IV-C2. Two examples are shown: *Chen* with a cyberware texture and *Luis_IST* with a frontal texture. The adaptation and mapping results shown in Figs. 15 and 16 also include the mapping results using the model projection not corresponding to the type of texture received. Fig. 17 shows some animation results for *Chen* using the sequence *Expressions*.

The obtained results indicate that good adaptation and texture-mapping results may be obtained if a cyberware-type texture is sent. However, for many applications such as videotelephony, only a frontal texture of the face will very likely be available, and this may not be enough for a good texture mapping (as well as for feature-point extraction). For these cases, a possible solution could be to use the MPEG-4 OCI stream to send the receiver the identification of the person calling; the receiver would then look in his terminal for an adapted model (with texture) of the person in question (if available). The results also show that having previous knowledge of which type of texture is being used allows better mapping and animation results. Regardless, the major conclusion is that texture mapping is a critical process in terms of the subjective impact of the animations, improving or decreasing it substantially if the mapping is good or bad. Unless good texture mapping can be achieved, it is better just to use a simple shadowing of the model since the user expectations and critical sense seem to increase with the realism of the animation.

3) Model Adaptation with FAT: The FAT tool can give the sender full control of the animation results at the receiver.

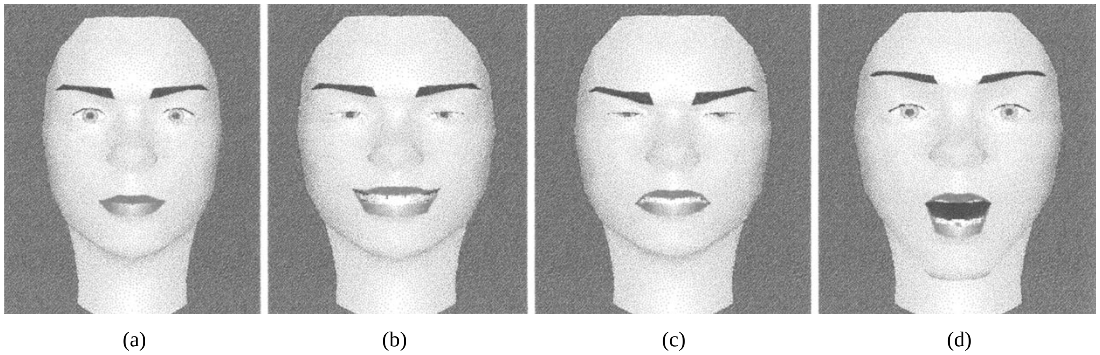


Fig. 18. *Baldy.* (a) Neutral face. Model animated with the sequence *Expressions:* (b) joy, (c) anger, and (d) surprise.

Fig. 18 shows the neutral model transmitted to the receiver, as well as a few frames of the same model animated with the sequence *Expressions*, using the FAT sent with the model for the test case *Baldy*. The same model can be animated with the other FAP sequences available, using the sender's defined FAT. The results shown in Fig. 18 may be compared with those in Fig. 11, since the same interpretation is given to the expressions in terms of sets of FAP's (but now the FAP interpretation models are different).

The FAT tool is the only one that can guarantee the sender a high level of fidelity for the animation results. The experience gained with the IST facial animation system supports the statement that model adaptation by means of FAT with a new 3-D model is not necessarily more complex than the model adaptation solutions using feature points, which may become quite complex (and sometimes with disappointing results). A large part of the required decoder memory and computational power will depend on the complexity of the model transmitted, notably the number of vertices, as well as on the complexity of the FAT. If the initial delay is not a problem, transmitting a FAT with a new model may be much simpler and effective than trying to adapt the receiver's model.

VI. Final Remarks

After many years of work on facial animation technology, MPEG-4 decided to standardize the minimum elements necessary to support a new range of applications characterized by the integration of 3-D faces. These 3-D faces are just another of the natural and synthetic data types allowed in the context of the MPEG-4 natural and synthetic representation framework.

To reach these objectives, MPEG-4 decided to standardize two main types of facial animation information: the facial animation parameters addressing the animation of the model and the facial definition parameters allowing model adaptation. The MPEG-4 facial animation solution claims model independence, providing complete freedom in terms of the 3-D facial model to be used. To support facial animation decoders with different degrees of complexity, MPEG-4 uses a profiling strategy that, for the moment, foresees the specification of one facial animation object type and three visual profiles including this object type.

This paper described the implementation of an MPEG-4 facial animation system, including the normative and non-normative parts, considering all the facial animation tools but FIT. Moreover, it presented a first evaluation of the performance of the various tools and corresponding object types. The facial animation system presented here has been implemented in parallel with the development of the MPEG-4 specification, allowing the authors to actively contribute to its improvement and completeness. Although a standard does not necessarily have to include the optimum technology since many constraints have to be compromised, it needs to be timely, clear, and precise. In fact, a standard will always have a certain lifetime, after which a new and better technology should lead to a new standard. A good standard is thus the standard that for a certain period of time allows one to make products, hardware or software based, providing users with solutions for the tasks that they have to perform or just entertaining them, with the biggest possible degree of interoperability for the lowest price. None of these objectives can be accomplish with a poor standard specification, even if the technology behind it is intrinsically good and adequate.

Let us finally hope that many new and improved applications including MPEG-4 3-D animated faces will appear in the near future. This would be our reward.

Acknowledgment

The authors wish to acknowledge the MPEG members who made the test material available.

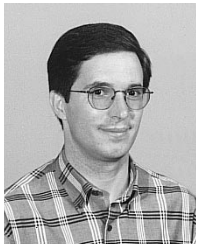
References

[1] Moving Picture Experts Group. [Online]. Available WWW: http://www.cselt.it/mpeg.
[2] R. Koenen, F. Pereira, and L. Chiariglione, "MPEG-4: Context and objectives," *Image Commun. J.*, vol. 9, no. 4, pp. 295–304, May 1997, http://drogo.cselt.it/ufv/leonardo/icjfiles/mpeg-4_si/paper1.htm.
[3] MPEG Systems, "Text of ISO/IEC FDIS 14496-1: Systems," Doc. ISO/MPEG N2501 Atlantic City MPEG Meeting, Oct. 1998.
[4] MPEG Video and SNHC, "Text of ISO/IEC FDIS 14496-2: Visual," Doc. ISO/MPEG N2502, Atlantic City MPEG Meeting, Oct. 1998.
[5] MPEG Audio, "Text of ISO/IEC FDIS 14496-3: Audio," Doc. ISO/MPEG N2503, Atlantic City MPEG Meeting, Oct. 1998.
[6] MPEG, "MPEG-4 Version 2 overview," Doc. ISO/MPEG N2324, Dublin MPEG Meeting, July 1998.
[7] G. Abrantes and F. Pereira, "Interactive analysis for MPEG-4 facial models configuration," in *EUROGRAPHICS'98–Short Presentations*, Lisbon, Portugal, Sept. 1998, pp. 1.6.1–1.6.4.
[8] F. Parke, "Parameterized models for facial animation," *IEEE Comput. Graph. Appl. Mag.*, vol. 2, pp. 61–68, Nov. 1982.
[9] MPEG, "Final text for FCD 14496-5: Reference software," Doc. ISO/MPEG N2205, Tokyo MPEG Meeting, Mar. 1998.

[10] G. Abrantes, "Analysis and synthesis of human faces using 3D models," M.Sc. thesis, Instituto Superior Técnico, Lisboa, Portugal, July 1998 (in Portuguese).

Gabriel Antunes Abrantes (S'98) was born in Heilbronn, Germany, in March 1973. He received the bachelor's and M.Sc. degrees in electrical and computer engineering from the Instituto Superior Técnico (IST), Universidade Técnica de Lisboa, Portugal, in 1996 and 1998, respectively.

Since 1996, he has been a Member of the Instituto de Telecomunicacões, Lisboa. His research interests are in the area of facial animation and facial analysis for coding. He participated in the activities that led to the MPEG-4 facial animation specification. He developed an MPEG-4 facial animation system for the European project ACTS MoMuSys. In 1998, he was a Research Consultant in the field of image processing with AT&T Laboratories, Red Bank, NJ.



Fernando Pereira (S'88–M'90) was born in Vermelha, Portugal, in October 1962. He received the bachelor's, M.Sc., and Ph.D. degrees in electrical and computer engineering from the Instituto Superior Técnico (IST), Universidade Técnica de Lisboa, Portugal, in 1985, 1988, and 1991, respectively.

He currently is a Professor in the Electrical and Computer Engineering Department at IST. He is responsible for the participation of IST in many national and international research projects. He often acts as Project Evaluator and Auditor at the invitation of the European Commission. He is a member of the editorial board of *Signal Processing: Image Communication*. He is a member of the Scientific Committee of many international conferences. He has written more than 80 papers. He has participated in the work of ISO/MPEG for many years, notably as the Head of the Portuguese delegation, and has chaired many ad hoc groups related to the MPEG-4 and MPEG-7 standards. His current areas of interest are video analysis, processing, description and representation, and multimedia interactive services.

Dr. Pereira is an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY.