

VERY LOW BITRATE CODING OF VIRTUAL HUMAN ANIMATION IN MPEG-4

Tolga K. Capin¹, Eric Petajan², Joern Ostermann³

¹Computer Graphics Laboratory
Swiss Federal Institute of
Technology (EPFL)
1015 Lausanne, Switzerland
capin@lig.di.epfl.ch

²face2face animation, inc.
2 Kent Place Blvd.
Summit, NJ 07901
eric@f2f-inc.com

³ AT&T Labs - Research
Rm 3-231
100 Schultz Dr
Red Bank, NJ 07701
osterman@research.att.com

ABSTRACT

This and the accompanying paper present an overview of the face and body animation object in the MPEG-4 Version 2 standard. The MPEG-4 standard includes the representation, compression of virtual human models and their interface with other MPEG-4 objects, with bit rate requirements as low as 1 Kbit/second. In this paper, we give an overview of animation techniques for the FBA object, and in the accompanying paper we describe the geometrical modeling of virtual human models.

1. INTRODUCTION

The MPEG-4 Version 2 standard defines a Face and Body Animation object, with the goal to define synthetic human face and body models with animations [4][5]. The FBA object is a collection of nodes in a scene graph, which are animated by the FBA (Face and Body Animation) object bitstream. Two separate bit streams control the FBA object. The first bitstream, called BIFS, contains instances of Body Definition Parameters (BDPs) in addition to Facial Definition Parameters (FDPs) [1], and the second bitstream, FBA elementary bitstream, contains Body Animation Parameters (BAPs) and Facial Animation Parameters (FAPs).

A 3D (or 2D) *face* is a representation of the human face that is structured for portraying the visual manifestations of speech and facial expressions adequate to achieve visual speech intelligibility and the recognition of the mood of the speaker. The face is animated by a set of *face animation parameters (FAP)*, each manipulating key feature control points in a mesh model of the face to produce animated visemes for the mouth (lips, tongue, teeth), as well as animation of the head and facial features such as the eyes.

A *body model* is a representation of a virtual human or human-like character that allows portraying body movements adequate to achieve nonverbal communication and general actions. A body model is animated by a stream of *body animation parameters (BAP)*. The BAPs manipulate independent degrees of freedom in the skeleton model of the body to produce animation of the body parts.

The FAPs and BAPs, if correctly interpreted, will produce reasonably similar high-level results in terms of facial expressions and body postures on different FBA models, also

without the need to initialize or calibrate the model. The FDP/BDP set defines the set of parameters to transform the default face and body to a customized model optionally with its surface, dimensions, and texture.

The FAPs and BAPs are encoded for low-bitrate transmission in broadcast and dedicated interactive communications. They are quantized with careful consideration for the limited movements of facial features and body parts, and then prediction errors are calculated and coded arithmetically. The remote manipulation of an FBA model in a terminal with FAPs and BAPs can accomplish lifelike visual scenes of a person in real-time without sending pictorial or video details of imagery every frame.

2. FBA OBJECT IN MPEG-4

The MPEG-4 The goal for defining the FBA object is to specify sufficient parameters for animating both realistic and cartoon-like humanoid characters. There is no constraint on the complexity of the models: the models can be realistic representations of real persons, as well as very simple cartoon-like models. To achieve this goal, MPEG-4 defines a large set of parameters to animate the FBA object in real time, and the applications can select subsets of these parameters to animate less complex models.

The specification of the FBA object is distributed into two separate but related parts of the MPEG-4 specification: Systems (Part 1) and Visual (Part 2). The Systems part specifies representation and coding of the geometry and method to deform the surface of face/body, i.e. FDP/BDP parameters. The Visual part specifies the coding of animation parameters, i.e. FAP/BAP parameters, and integration of the FBA node with other objects e.g. text-to-speech streams. This paper discusses the visual part; the accompanying paper discusses the systems part [1].

The MPEG-4 FBA visual specification defines the syntax of the bit stream and the decoders' behavior. The way of generating the motion (e.g. visual tracking, key frame animation and autonomous agent animation) is not specified by the standard, the encoders are free to choose the method depending on their application. The standard defines a scalable coding scheme, and the encoder can choose among coding parameters to achieve a selected bitrate and animation quality.

2.1 FAPs and BAPs

As discussed in [1], Face Definition Parameter (FDP) feature points have been defined and located on the face. Some of these points only serve to help define the shape of the face. The rest of them are displaced by FAPs that are listed in the standard. FAPs 1 and 2 are sets of descriptors for visemes and expressions respectively. The remaining FAPs (except for the rotation FAPs) are normalized to be proportional to one of neutral face mouth width, mouth-nose distance, eye separation, iris diameter, or eye-nose distance.

FAPs are displacements of the feature points from the neutral face position. Neutral position is defined as mouth closed, eyelids tangent to the iris, gaze and head orientation straight ahead, teeth touching, and tongue touching teeth [1].

FAPs, which are not transmitted for a given frame, may be interpolated by the decoder. For example, if only the inner lip and not the outer lips FAPs are transmitted, the decoder is free to synthesize the motion of the outer lips. Typically, the outer lip motion would closely follow the motion of the inner lips.

Table 1: FAP groups

Group	Number of FAPs
1: visemes and expressions	2
2: jaw, chin, inner lowerlip, cornerlips, midlip	16
3: eyeballs, pupils, eyelids	12
4: eyebrow	8
5: cheeks	4
6: tongue	5
7: head rotation	3
8: outer lip positions	10
9: nose	4
10: ears	4

BAPs manipulate independent degrees of freedom in the skeleton model of the body to produce animation of the body parts. Similar to the face, the remote manipulation of a body model in a terminal with BAPs can accomplish lifelike visual scenes of the body in real-time without sending pictorial and video details of the body every frame.

The BAPs, if correctly interpreted, will produce reasonably similar high level results in terms of body posture and animation on different body models, also without the need to initialize or calibrate the model.

There are a total of 186 predefined BAPs in the BAP set, with an additional set of 110 user-defined extension BAPs. Each predefined BAP corresponds to a degree of freedom in a joint connecting two body parts. These joints include toe, ankle, knee, hip, spine (C1-C7, T1-T12, L1-L5), shoulder, clavicle, elbow, wrist, and the hand fingers. Extension BAPs are provided to animate additional features than the standard ones in connection with body deformation tables [1], e.g. for cloth animation.

The BAPs are categorized into groups with respect to their effect on the body posture. Using this grouping scheme has a

number of advantages. First, it allows us to adjust the complexity of the animation by choosing a subset of BAPs. For example, the total number of BAPs in the spine is 72, but significantly simpler models can be used by choosing only Spine1 group. Secondly, assuming that not all the motions contain all the BAPs, only the active BAPs can be transmitted to decrease required bit rate significantly. This is accomplished by using a mask transmitted with the active BAP groups in a frame as discussed in Section 3.

Table 2: BAP groups

Group	Number of BAPs
1. Pelvis	3
2. Left leg1	4
3. Right leg1	4
4. Left leg2	6
5. Right leg2	6
6. Left arm1	5
7. Right arm1	5
8. Left arm2	7
9. Right arm2	7
10. Spine1	12
11. Spine2	15
12. Spine3	18
13. Spine4	18
14. Spine5	12
15. Left hand1	16
16. Right hand1	16
17. Left hand2	13
18. Right hand2	13
19. Global positioning	6
20. Extension BAPs1	22
21. Extension BAPs2	22
22. Extension BAPs3	22
23. Extension BAPs4	22
24. Extension BAPs5	22

2.2 FBA AND TEXT-TO-SPEECH

MPEG-4 acknowledges the importance of text-to-speech (TTS) synthesis for multimedia applications providing an interface to proprietary text-to-speech synthesizer (TTSI). A TTS stream contains text in ASCII and optional prosody in binary form. The decoder decodes the text and prosody information according to the interface defined for the TTS synthesizer. The synthesizer creates speech samples that are handed to the compositor. The compositor presents audio and if required video to the user.

In the current MPEG4 standard, the encoder is expected to send a FAP stream containing FAP number and amplitude for every frame, to enable the receiver to produce desired facial actions (Figure 1). Since the TTS synthesizer can behave like an asynchronous source, synchronization of speech parameters with facial expressions of the FAP stream is usually not given – unless the encoder transmits prosody with timing information for the synthesizer.

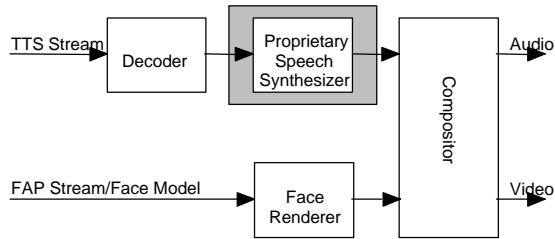


Figure 1: Block diagram showing the integration of a proprietary Text-to-Speech Synthesizer into an MPEG-4 face animation system.

Figure 2 shows the architecture of the visual TTS architecture that allows synchronized presentation of synthetic speech and talking heads. A second output interface is added to the TTS. This interface sends the phonemes of the synthesized speech as well as start time and duration information for each phoneme to a Phoneme/Bookmark-to-FAP-Converter. The converter translates the phonemes and timing information into face animation parameters that the face renderer uses in order to animate the face model. In addition to the phonemes, the synthesizer identifies bookmarks in the text that convey non-speech related facial animation parameters to the face renderer. The timing information of the bookmarks is derived from their position in the synthesized speech. Since now the facial animation is driven completely from the text input to the TTS, transmitting an FAP stream to the decoder is optional. Furthermore, synchronization is achieved since the talking head is driven by the speed of the asynchronous proprietary TTS synthesizer.

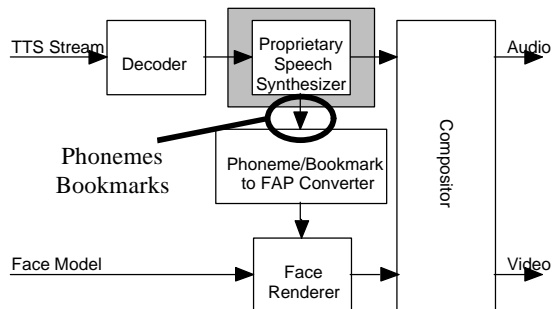


Figure 2: Architecture for VTTS allowing synchronization of facial expressions and speech.

In order to allow for simple bookmarks, each bookmark has to describe for one FAP at a time the transition from the current FAP amplitude to a target FAP amplitude. Simply applying an FAP of constant amplitude and resetting it after a certain amount of time does not allow for realistic face motion. Therefore, the Bookmark to FAP Converter creates the appropriate transition between current amplitude and the target amplitude. There are 2 ways of designing bookmarks:

1. The position of the bookmark defines the amplitude of the FAP at the time instant of the spoken word. Consequence: In order to generate smooth temporal behavior of the FAP the decoder has to look *ahead* into the TTS stream in order to determine an appropriate behavior. This increases the delay of the decoder.

2. The bookmark defines the *start* point and *duration* of the transition to a new FAP amplitude. Consequence: No additional delay, no look ahead in the bitstream but no precise timing control on when the amplitude will be reached relative to the spoken text.

In our tests we did not find a problem with using option 2 since the transition times for facial expressions is usually less than 1s. As syntax for a bookmark, we use $\langle \text{FAP } n \text{ (s) } a \text{ } T \rangle$ with FAP number n , expression s in case n equals 2, the amplitude a and the transition time T in ms.

3. FBA CODING

3.1 FAP/BAP Coding

MPEG-4 provides two alternative tools for coding FAPs and BAPs. Coding of quantized and temporally predicted FAPs/BAPs using an arithmetic coder permits low-delay FAP coding. Alternatively, discrete cosine transform (DCT) coding in the temporal direction of a sequence of values of FAPs/BAPs in a group of frames introduces a larger delay but achieves higher coding efficiency.

Arithmetic coding of FBA frames is applied as follows: at an I-frame, FAPs/BAPs are coded without prediction. At P-frames following this frame, the values of FAPs and BAPs at frame k $\text{FAP}_k/\text{BAP}_k$, are predicted using the previously decoded value $\text{FAP}_{k-1}/\text{BAP}_{k-1}$. The prediction error is quantized using a quantization step size QP different for each FAP/BAP, multiplied by a global quantization parameter FQU/BQU . The quantization step sizes QP for each FAP and BAP have been chosen based on their accuracy requirements and experimental results.

The second coding scheme, DCT coding, is applied to 16 consecutive FBA frames. This introduces a delay into coding and decoding process, hence this process is useful for applications where FBA frames are retrieved from a file. After computing the DCT of 16 consecutive value for one FAP/BAP, DC and AC coefficients are coded differently. Whereas the DC value is coded predictively using the previous DC coefficient as prediction, the AC coefficients are directly coded. The AC coefficient and the prediction error of the DC coefficient are linearly quantized.

Figure 3 demonstrates the syntax of the FBA elementary stream that contains predictive coded FAPs and BAPs. Note that this coding scheme uses a masking technique. This is based on the observation that not all FAPs and BAPs will be present in a frame. There are different levels of masking. The first masking appears in the object level: a frame can contain either face or body, or both. The second level of masking occurs in the FAP or BAP group level. If the model is a simple model with some of the FAP/BAP groups omitted (e.g. a simple model with 9-dof spine), then the Face or Body Group Mask Type can be set to 0 for that group. Furthermore, the third level of masking is used within an individual FAP/BAP group. If the mask type for an individual group is set to '01', then a further group mask follows the mask type. This group mask is a sequence of bits,

with the size of FAPs/BAPs in that group. Each bit in the mask specifies if that FAP/BAP is present in the current frame. Following this variable-length mask, the FAPs and BAPs in this frame are transmitted are coded using arithmetic coding [5].

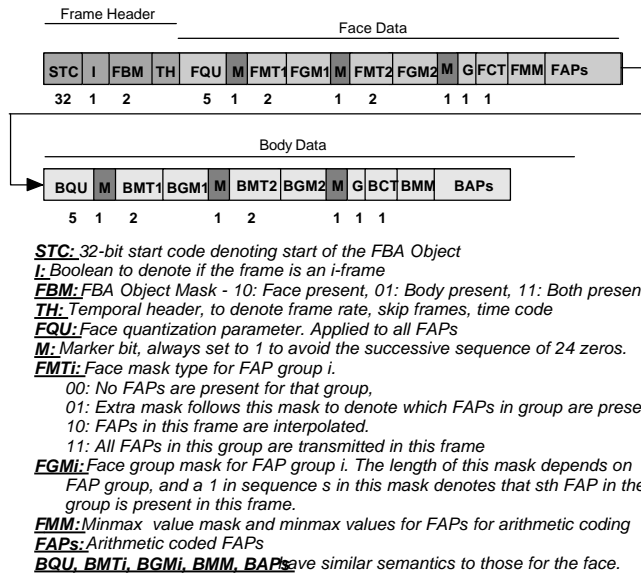


Figure 3: FBA visual syntax for predictive coding

4. EXPERIMENTAL RESULTS

We have performed experiments on coding FBA sequences, each containing several hundreds of frames. Table 3 shows the the coded FAP and BAP bitrate as a function of the quantization parameter, using frame-based coding. The FAP sequence includes lip movements and head rotation at 30 frames per second, and the BAP sequence includes body movements for animation of an online presenter, created using keyframe animation, at also 30 frames per second. These sequences were chosen to represent common animation characteristics.

Table 3 demonstrates several results. Column 2 shows the bitrate requirements for face animation only, coded for transmission of only updated FAPs at every frame. Columns 3 and 4 show the transmission of all predefined BAPs, using only I-frames, and first I-frame followed by all P-frames, respectively. Column 5 gives the bit rate requirements for body animation, coded for transmission of only updated BAPs at every frame. Finally, the last column shows the total bit rate for coding FAPs and BAPs in the FBA bitstream.

The second column of the table shows that as little as 0.3 Kbits/s are required for face animation at 30 fps. Secondly, comparison of columns 3 and 4 demonstrate that using P-frames instead of I-frames decreases the required bit rate significantly. Thirdly, the drastic decrease of bit rate required by coding updated BAPs (column 5), with respect to all the BAPs (column 4) shows that only a few (approximately 10%) of all the BAPs are active in the body for this animation. Finally, the column 5 shows that the FBA object requires 4.4 Kbits/sec for good quality animations, and 1 Kbits/sec for lower quality results, at 30 fps.

Table 3: Bit rate as a function of FAP/BAP quantization for a frame rate of 30 fps, in Kbits/sec

Quant Step size	Face Only, Efficient coding	Body Only			Both face and body
		All I-frames	First I-Frame, all P-frames	Periodic I-frames, Efficient coding	
1	2.295	45.000	19.000	2.183	4.478
2	1.800	37.000	12.000	1.961	3.761
4	1.405	30.000	7.000	1.669	3.070
8	0.974	22.000	4.000	1.345	2.319
16	0.551	18.000	3.000	1.009	1.560
31	0.300	14.000	2.000	0.705	1.001

5. CONCLUSIONS

In this paper, we have presented the visual part of the Face and Body Animation coding in MPEG-4, for very low bitrate compression of human-like characters. The methods presented give efficient compression and are suitable for real-time communication applications such as videoconferencing, e-commerce, games, multi-user chat worlds.

Further additions to the techniques will be perhaps in content creation side. Choosing the encoding parameters to suit a given quality of service and error value is still an area of research. Furthermore, the extension BAPs could be used to define high-level animations, such as nonverbal communication. Additionally, using FBA models as avatars in multi-user worlds might require techniques to lower required bitrate per avatar, using high-level or dead-reckoning techniques.

6. REFERENCES

- [1] Capin Tolga, Eric Petajan, Joern Ostermann, Efficient Modeling of Virtual Humans in MPEG-4, *Proc. ICME'2000*, New York, NY, July 2000.
- [2] Web3D Working Group on Humanoid Animation, Specification for a Standard Humanoid, Version 1.1, August 1999.
- [3] MPEG-4 Overview, ISO/IEC JTC1/SC29 N2995, available at <http://drogo.cselt.it/mpeg/standards/mpeg-4/mpeg-4.htm>, October 1999.
- [4] ISO/IEC 14496-1:1999, Coding of Audio-Visual Objects: Systems, Amendment 1, December 1999.
- [5] ISO/IEC 14496-2:1999, Coding of Audio-Visual Objects, Visual, Amendment 1, December 1999.
- [6] Signes Julien, Yuval Fischer, Alexandros Eleftheriadis, MPEG-4's Binary Format for Scene Description, *Signal Processing: Image Communication*, Vol. 15, No.4-5, pp 321-345, January 2000.
- [7] VRML97: ISO/IEC 14772-1:1997, The Virtual Reality Modeling Language, 1997.