INTEGRATION OF TALKING HEADS AND TEXT-TO-SPEECH SYNTHESIZERS FOR VISUAL TTS

Jörn Ostermann[1], Mark Beutnagel[1], Ariel Fischer[2], Yao Wang[3]

[1]AT&T Labs Research, [2]Institute Eurecom/EPFL, [3]Polytechnic University

ABSTRACT

The integration of text-to-speech (TTS) synthesis and animation of synthetic faces allows new applications like visual human computer interfaces using agents or avatars. The TTS informs the talking head when phonemes are spoken. The appropriate mouth shapes are animated and rendered while the TTS produces the sound. We call this integrated system of TTS and animation a Visual TTS (VTTS). This paper describes the architecture on an integrated VTTS synthesizer that allows defining facial expressions as bookmarks in the text that will be animated while the model is talking.

The position of a bookmark in the text defines the start time for the facial expression. The bookmark itself names the expression, its amplitude and the duration during which the amplitude has to be reached by the face. A bookmark to face animation parameter (FAP) converter creates a curve defining the amplitude for the given FAP over time using Hermite functions of 3rd order.

1. INTRODUCTION

With the new generation of Text-to-Speech (TTS) synthesizers creating human like and pleasant voices, the integration of text-to-speech (TTS) synthesis and animation of synthetic faces allows new applications like visual human computer interfaces (HCI), information kiosks or networked applications like virtual sales agents and virtual company representatives. We name such a system Visual TTS (VTTS). In a VTTS system, the speech synthesizer informs the face renderer about the phonemes of the spoken text and the related timing information. A phoneme to viseme converter computes appropriate mouth shapes that generate the impression of a talking head when rendered while the TTS produces the sound. TTS and face animation are areas covered by the upcoming MPEG-4 standard.

The goal of MPEG-4 is to provide a new kind of standardization that responds to the evolution of technology, when it does not always make sense to specify a rigid standard addressing just one application. MPEG-4 will allow the user to configure and build systems for many applications by allowing flexibility in the system configurations, by providing various levels of interactivity with audio-visual content of a scene, and by integrating audio visual data types like natural and synthetic audio, video and graphics [1][2][3]. MPEG-4 will become an International Standard in spring 1999, just in time for the new faster and more powerful media processors and in time for using the upcoming narrow- and broadband wired and wireless networks for audio-visual applications like database browsing, information retrieval and interactive communications.

As far as synthetic multimedia contents are concerned, MPEG-4 will provide synthetic audio like structured audio and a text-to-speech interface (TTSI). For synthetic visual contents, MPEG-4 allows to build 2D and 3D objects composed of primitives like rectangles, spheres, indexed facesets and arbitrarily shaped 2D objects. The 3D-object description is based on a subset of VRML nodes [4] and extended to enable seamless integration of 2D and 3D objects. Objects can be composed into 2D and 3D scenes using the BInary Format for Scenes (BIFS). BIFS also allows to animate objects and their properties.

Special 3D objects are human faces and bodies. MPEG-4 allows using decoder resident proprietary models as well as transmission of 3D models to the decoder such that the encoder can predict the quality of the presentation at the decoder [3]. The integration of TTS and facial animation is currently limited. Non-speech related animation parameters are transmitted using a synchronous stream. Since the timing of the TTS is unknown to the sender, synchronization of facial expressions with mouth shapes and sound cannot be achieved. Here, we propose an architecture that allows driving a face model, including its facial expressions, based on MPEG-4 face animation parameters (FAP) from the text input of the TTS using a bookmark mechanism and an interpolation function to derive the amplitude of the facial expressions over time given the bookmarks.

In Sections 2, we explain how MPEG-4 defines the specification of a face model and its animation using FAPs. Section 3 shows the proposed architecture to combine face animation with text-to-speech capabilities. Section 4 describes different interpolation functions for FAP amplitudes.

2. FACE ANIMATION IN MPEG-4

MPEG4 specifies a set of face animation parameters (FAPs), each corresponding to a particular facial action deforming a face model in its neutral state. The FAP value for a particular FAP indicates the magnitude of the corresponding action, e.g., a big versus a small smile. Deforming the face model in its neutral state according to the specified FAP values for the corresponding time instant generates a particular facial action sequence. Then the model is rendered onto the screen.
The head in its neutral state is defined as follows (Figure 1): Gaze is in direction of Z axis; all face muscles are relaxed; eyelids are tangent to the iris; the pupil is one third of IRISD0; lips are in contact; the line of the lips is horizontal and at the same height of lip corners; the mouth is closed and the upper teeth touch the lower ones; the tongue is flat, horizontal with the tip of tongue touching the boundary between upper and lower teeth.

For the renderer to interpret the FAP values using its face model, the renderer has to have predefined model specific animation rules to produce the facial action corresponding to each

FAP. Since the FAPs are required to animate faces of different sizes and proportions, the FAP values are defined in face animation parameter units (FAPU). FAPU are defined as fractions of distances between key facial features (Figure 1). These features like eye separation, eye-nose separation, mouth nose separation, and mouth width, are defined for the face in its neutral state. They allow interpretation of the FAPs on any facial model in a consistent way, producing reasonable results in terms of expression and speech pronunciation.

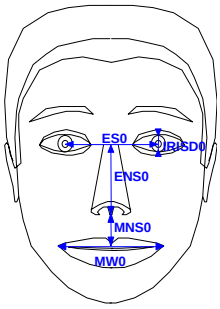

Figure 1: FAPUs [2].

MPEG-4 defines the animation rule for each FAP by specifying feature points and their direction of movement. The renderer can either use its own animation rules for its proprietary model or download a face model and its FaceDefTables that define the animation rules for the model.

2.1. Face Feature Points

In order to define face animation parameters for arbitrary face models, MPEG-4 specifies 84 feature points located in a face according to Figure 2 in order to provide a reference for defining facial animation parameters. The location of these feature points has to be known for any MPEG-4 compliant face model.

2.2. Face Animation Parameters

The FAPs are based on the study of minimal perceptible actions and are closely related to muscle action [5]. The 68 parameters are categorized into 10 groups related to parts of the face. FAPs represent a complete set of basic facial actions including head motion, tongue, eye, and mouth control. They allow the representation of natural facial expressions. They can also be used to define facial action units [6]. Exaggerated values permit the definition of actions that are normally not possible for humans, but are desirable for cartoon-like characters.

The FAP set contains two high-level parameters: visemes and expressions. A viseme is a visual correlate to a phoneme. Only 14 static visemes that are clearly distinguished are included in the standard set (Table 1). The expression parameter defines 6 high level facial expressions like joy and sadness (Figure 3). In contrast to visemes, facial expressions are animated with a value defining the amplitude of the expression. Two facial expressions can be blended with a weighting factor. Since expressions are high-level animation parameters, they allow animating unknown models with high subjective quality.

2.3. Face Model Specification

MPEG-4 allows the encoder to completely specify the face model the decoder has to animate. This involves defining the static geometry of the face model in its neutral state using a scene graph and defining the animation rules that specify how this model gets deformed by the facial animation parameters [8].

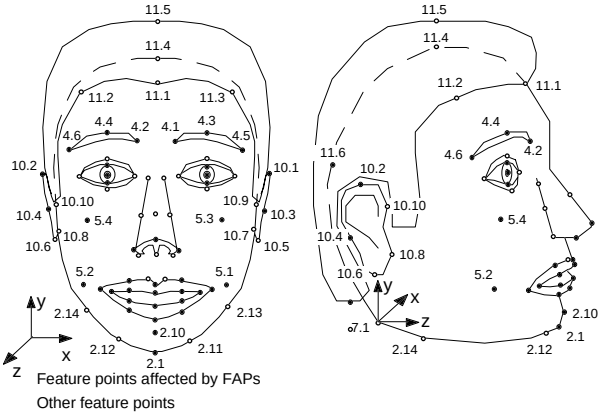

Figure 2: Subset of MPEG-4 face feature points used to define the shape of a proprietary face model. FAPs are defined by motion of feature points [2].

Table 1: Visemes and related phonemes.

#	phoneme	example	#	phoneme	example
1	p, b, m	put, bed, mill	8	n, l	lot, not
2	f, v	far, voice	9	r	red
3	T,D	think, that	10	A:	car
4	t, d	tip, doll	11	e	bed
5	k, g	call, gas	12	I	tip
6	tS,dZ,S	chair, join, she	13	Q	top
7	s, z	sir, zeal	14	U	book

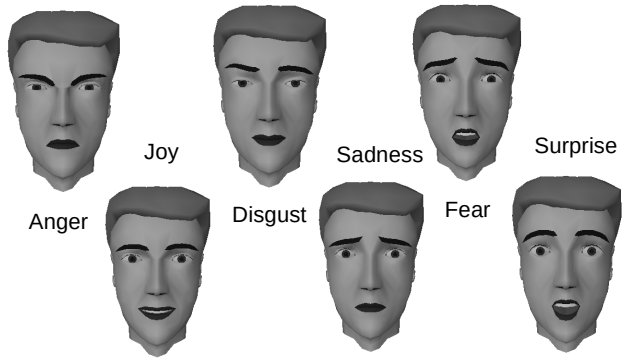

Figure 3: Primary facial expressions.

2.4. Integration with TTS

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 4). 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.

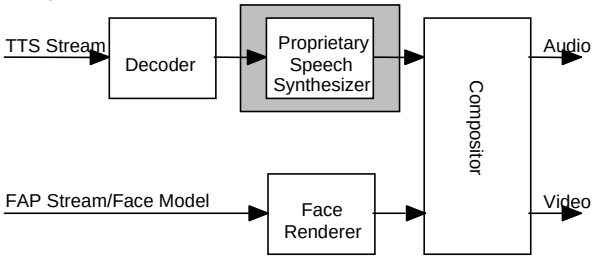

Figure 4: Block diagram showing the integration of a proprietary Text-to-Speech Synthesizer into an MPEG-4 face animation system.

3. ARCHITECTURE FOR VTTS

Figure 5 shows the architecture of the proposed VTTS 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 [7][9]. 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, there is no need to transmit an FAP stream to the decoder. Furthermore, synchronization is achieved since the talking head is driven by the speed of the asynchronous proprietary TTS synthesizer.

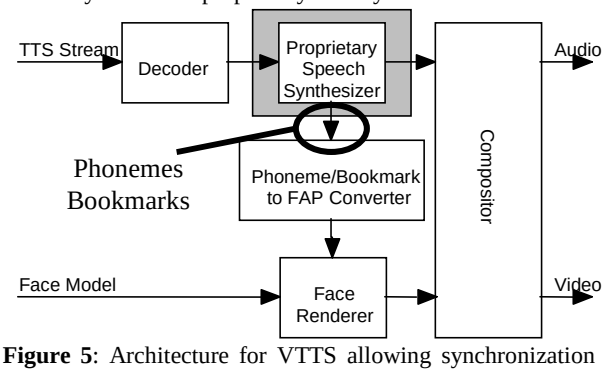

Figure 5: Architecture for VTTS allowing synchronization of facial expressions and speech.

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, we propose that 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 <FAP *n* (*s*) *a T*> with FAP number *n*, expression *s* in case *n* equals 2 (Figure 3), the amplitude *a* and the transition time *T* in ms.

4. INTERPOLATION FUNCTIONS

The FAP amplitude *a* defines the amplitude to be applied at the end of the transition time *T*. The amplitude a_s of the FAP at the beginning of the transition depends on previous bookmarks and can be equal to:

- 0 if the FAP bookmark is the first one with this FAP *n*.
- *a* of the previous FAP bookmark with the same FAP *n* if a time longer than the previous transition time *T* has elapsed between these two FAP bookmarks.
- The actual reached amplitude due to the previous FAP definition if a time shorter than the previous transition time *T* has elapsed between the two FAP bookmarks.

At the end of the transition time *T*, *a* is maintained until another FAP bookmark gives a new value to reach. To reset an FAP, a bookmark for FAP *n* with *a*=0 is transmitted in the text.

To avoid too many parameters for defining the evolution of the amplitude during the transition time, the function that computes for each frame the amplitude of the FAP to be sent to the face renderer is predefined. Assuming that the transition time *T* is always 1, we implemented the following functions *f(t)*:

$$f(t) = a_s + (a - a_s)t \quad (1)$$

$$f(t) = a_s + (1 - e^{-t})(a - a_s) \quad (2)$$

$$f(t) = a_s + \left(1 - e^{-\lambda(t-1/2)}\right)^{-1}(a - a_s) \quad (3)$$

$$f(t) = (2t^3 - 3t^2 + 1)a_s + (-2t^3 + 3t^2)a + (t^3 - 2t^2 + t)g_s \quad (4)$$

with time $t \in [0,1]$, the amplitude a_s at the beginning of the FAP at t=0, control parameter λ and the gradient g_s of $f(0)$ which is the FAP amplitude over time at t=0. If the transition time $T \neq 1$, the time axis of the functions (1) to (4) has to be scaled. These functions depend on a_s, g_s, *a* and *T*, and thus they are completely determined as soon as the FAP bookmark is known. After extensive subjective evaluations, it turns out that the Hermite function of third order (4) gives the best results, in terms of realistic behavior. Using Splines with more than one Hermite

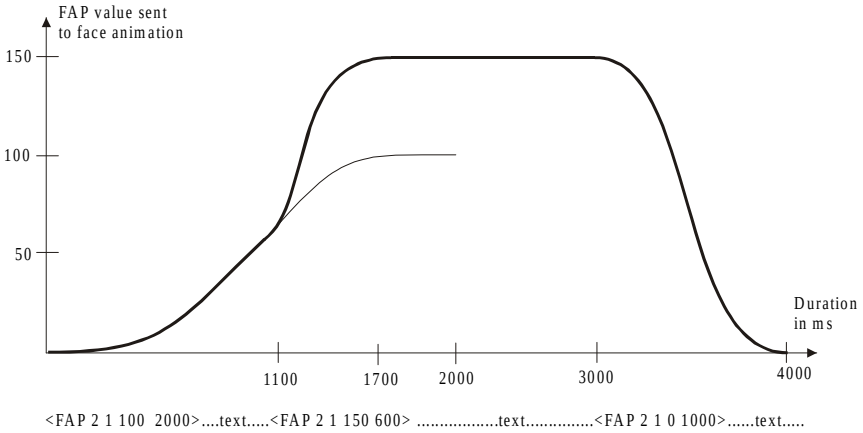

Figure 6: Amplitude of joy (FAP 2 1) as defined by the bookmarks.

segment would increase the flexibility for designing curves but would also require to have some knowledge of bookmarks placed further in the text than the current bookmark which is a significant drawback for a real-time system.

The Hermite function of third order enables one to match the tangent at the beginning of a segment with the tangent at the end of the previous segment, so that a smooth curve can be guaranteed. Usually, the computation of the Hermite function requires 4 parameters as input, which are a_s, g_s, a and the gradient of $f(t)$ at t=1. In our implementation we assume a horizontal gradient at the end of the transition time. Figure 6 shows an example of a time curve created with 3 bookmarks for FAP 2 (expression) and expression 1 (joy). As can be seen, the gradient g_s at the beginning of the transition time is 0 for the first and the third bookmark. The gradient g_s for the second bookmark at time t is computed according to

$$g_S(t) = (6t^2 - 6t)(a_S - a) + (3t^2 - 4t + 1)g_S \qquad (5)$$

with a_s, g_s ,and a defined by the first bookmark and g_s(t) defining the starting gradient g_s for the second bookmark.

5. Conclusions

MPEG-4 integrates animation of synthetic talking faces into audio-visual multimedia communications. A face model 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. A face model is defined as a static 3D model and related animation rules that define how the model deforms if it is animated with FAPs. The model is defined using a scene graph. Therefore, a customized model with head and shoulders can be defined for games or web-based customer service applications. MPEG-4 defines a complete set of animation parameters tailored towards animation of the human face. Face animation parameters are defined independent of the proportions of the animated face model. Therefore, a face animation parameter stream can be used to animate different models. Successful animations of humans, animals and cartoon characters have been demonstrated.

MPEG-4 defines interfaces to include a proprietary TTS synthesizer in an MPEG-4 multimedia application. However, MPEG-4 does not yet provide sufficient means to synchronize a face model with a TTS synthesizer.

Here, we propose an architecture that completely controls a talking head using the text input of the TTS. In addition to the input interface and the sound output interface as defined by MPEG-4 we define an interface that exports the phonemes and their timing from the TTS. In addition, the TTS is extended to be able to recognize bookmarks in the text. These bookmarks are also exported with their timing derived from the words between which they are located. A phoneme/bookmark converter translates the phonemes and bookmarks into appropriate sequences of facial animation parameters that are rendered such that speech and animation are synchronized.

One bookmark defines one facial expression, its amplitude and a transition time after which this amplitude has to be reached. Subjective evaluation showed that a Hermite function of 3rd order allows realistic animation of facial expressions.

This proposal for synchronizing TTS with facial expressions is currently considered by MEG-4.

6. References

[1] ISO/IEC JTC1/WG11 N2201, “Text for FCD 14496-1 Systems”, *Tokyo meeting, March 1998.*

[2] ISO/IEC JTC1/WG11 N2202, “Text for FCD 14496-2 Visual”, *Tokyo meeting, March 1998*.

[3] J. Ostermann, "Animation of synthetic faces in MPEG-4", *Computer Animation 98, pp. 49-55, Philadelphia, June 1998.*

[4] J. Hartman, J. Wernecke, *The VRML handbook*, Addison Wessley, 1996.

[5] Kalra P., Mangili A, Magnenat-Thalmann N, Thalmann D. “Simulation of Facial Muscle Actions Based on Rational Free Form Deformations”, *Proc. Eurographics 92, pp. 59-69, 1992*.

[6] P. Ekman, W.V. Friesen, *Manual for the facial action coding system*, Consulting Psychologist Press, Inc. Palo Alto, CA, 1978.

[7] M. M. Cohen and D. W. Massaro, “Modeling Coarticulation in Synthetic Visual Speech,” In M. Thalmann & D. Thalmann (Eds.) *Computer Animation ''93*, Tokyo: Springer-Verlag.

[8] J. Ostermann, E. Haratsch, “An animation definition interface: Rapid design of MPEG-4 compliant animated faces and bodies”, *International Workshop on synthetic - natural hybrid coding and three dimensional imaging, pp. 216-219, Rhodes, Greece, September 5-9, 1997*.

[9] K. Waters, T. Levergood, “An automatic lip-synchronization algorithm for synthetic faces”, *Proceedings of the Multimedia Conference, ACM, pages 149-156, San Francisco, California, September 1994*.