Text2Video: Text-Driven Facial Animation using MPEG-4

J. Rurainsky and P. Eisert

Fraunhofer Institute for Telecommunications - Heinrich-Hertz Institute
Image Processing Department
D-10587 Berlin, Germany
{rurainsky,eisert}@hhi.fraunhofer.de

ABSTRACT

We present a complete system for the automatic creation of talking head video sequences from text messages. Our system converts the text into MPEG-4 Facial Animation Parameters and synthetic voice. A user selected 3D character will perform lip movements synchronized to the speech data. The 3D models created from a single image vary from realistic people to cartoon characters. A voice selection for different languages and gender as well as a pitch shift component enables a personalization of the animation. The animation can be shown on different displays and devices ranging from 3GPP players on mobile phones to real-time 3D render engines. Therefore, our system can be used in mobile communication for the conversion of regular SMS messages to MMS animations.

Keywords: MPEG-4, Facial Animation, Text-Driven Animation, SMS, MMS

1. INTRODUCTION

Inter human and human-machine communication are two of the major defiances of this century. Video communication between people becomes more and more desirable with the fast growing available connectivity. The latest video compression techniques like H.264/AVC[1] or Windows Media 10 are able to highly reduce bit-rate for video data and enable communication over a wide variety of different channels. For even lower bandwidth, MPEG-4 standardized a communication system with 3D character models that are animated according to a set of Facial Animation Parameters (FAP).[2] These parameters describe motion and facial mimic of a person and can be efficiently encoded. Model-based video codecs based on MPEG-4 FAPs enable video communication at a few kbps.[3,4]

However, talking head videos can also be created without sending any information from a camera. Since there is a high correlation between the text, speech, and lip movements of the person, an artificial video can be synthesized purely from the text. We have developed a scheme, which allows the user to communicate by means of video messages created from the transmitted text. A Text-To-Speech engine (TTS) converts the message into a speech signal for different languages and markup information, like phonemes, phoneme durations, as well as stress levels for each phoneme. These side information are used to estimate MPEG-4 Facial Animation Parameters that are applied onto the 3D head model. Rendering of the head model leads to a realistic facial animation synchronized with the speech signal.[5–7]

A similar system, but with the extraction of the MPEG-4 FAPs on the receiver side is described in.[8] Realistic voices for TTS engines require a large set of speech samples, which have to be stored locally. The usage of a TTS engine for devices like PDAs and cellular phones requires either more memory than regular provided or the acceptance of less quality of the synthetic voice. Human-Machine Interfaces with a TTS engine for face animation are developed as well.[9,10]

In the following sections, we describe how our system requests input data as well as selections in order to individualize the video clips for different applications. Further, we describe the facial animation parameter creation and the rendering on different devices. We included a section for the transport within LAN and WLAN connections and explain different display interfaces.

DRAFT December 17, 2004 -- DRAFT December 17, 20

2. SYSTEM DESCRIPTION

We describe a system for facial animation of 3D models, that converts written text into a video-audio animation. The user is able to select a character (3D model), who will speak the text. Emoticons added to the text enable to control the 3D model besides the lip movements and to personalize the message for the receiver. Different voices with different languages and genders can be selected. Our pitch shift adjusts the synthetic voice to the character and increase the variability of the system. From the user input, MPEG-4 FAPs are created which efficiently describe the animation. Dependend on the application, the video can be rendered on the output device or a server creates a video which is streamed to the client. A system overview is given in **Fig. 1**. It follows a brief description of the system before we explain each element of the system more in detail.

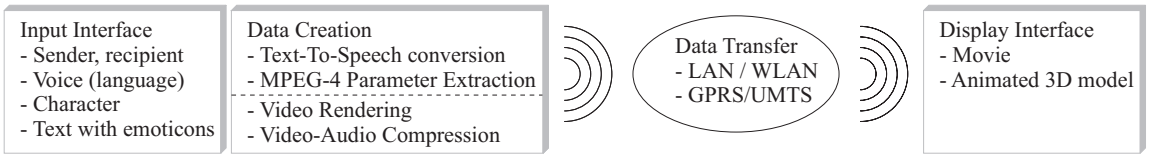

Figure 1. System overview as complete chain from the specified, data creation, transfer module, and finally the display interface.

Written text is the major input provided by the user. In addition to text, a set of emoticons (e.g. smile and sad) can be inserted between words and allow to personalize a message. A set of selections characterize the second major input. The user has to select a character from a list of already available 3D models. A gender and language selection allows to increase the field of usage and completes the set of required input data of our system. Predefined associations of gender to 3D face/head model and automated language detection are implemented as well.

The data creation step converts the written text into animation data for a 3D head model. First, a Text-To-Speech (TTS) system is used to convert the text into audio and phoneme data of the specified language. In a second step, MPEG-4 animation parameters are estimated from the provided phoneme data from the TTS. For low bandwidth connections, these parameters are streamed and real-time rendering is performed on the client. However, the videos can be created at the server also, which requires additional audio-video interleaving and compression. Our system currently supports MPEG-4 and 3GP formats.[11,12]

After creating the animation data, these data are transmitted to the receiver of this animated text message. Different channels from WLAN to GPRS are possible and already tested. Depending on the channel, the appropriate transport protocol or system has to be used. Our proprietary protocol is able to transmit different data to the receiver as a file transfer and is used for LAN or WLAN connection. GPRS transmissions to mobile phones are realized via a network provider. The final step of our system displays the created animation. Display types vary from real-time 3D render engines implemented on mobile devices to movie players on PCs and mobile phones.

The following sections describe all parts of our application in more detail as well as the interfaces between the individual parts.

2.1. Input Interface

The input interface requests the necessary information from the user/sender. The input data can be categorized into two types. First, arbitrary text input as well as the free usage of emoticons. Second, selections and side information like character, gender, and language.

The input text is specified in UTF-8 or ASCII format. The emoticons are represented by a character string. In **Tab. 1** a table is given, which shows the used mapping from either textual description or images of emoticons to the appropriate character strings. An example text could look like: "*Hello Mr. Smith, :-)) I am very happy about your message.*".

DRAFT December 17, 2004 -- DRAFT December 17, 20

Table 1. Mapping from textual description of image of emoticons to ASCII character strings.

textual description	ASCII character string
smile	:−)
laugh	:−))
sad	:−(
very sad	:−((
neutral	:−|

Selections are the second type of input, which allows the user/sender to control and individualize the output. A set of selections is given in **Tab. 2**. The selections *character* and *voice* are mandatory for all cases of usage or target display interfaces.

Table 2. Available selections to be made by the user/sender.

display option	example	textual description
character	woman 1 woman 2 man 1 cartoon robot 1	female persons movie actor former president non realistic figure robot
voice (male/female)	US english GB english French German	US pronunciation GB pronunciation
Sender Recipient	name, email, phone number	

In **Fig. 2** an example is given, which shows a World-Wide-Web based application. The user/sender first selects the desired service, which is either an instant display (left part of **Fig. 2**) or an eCard or eMail service (right part of **Fig. 2**). The desired language has to be selected in a second step or automatic language detection is chosen. Character and the text as shown in **Fig. 2** are the last required inputs. Depending on the service the user/sender has to provide contact information of the recipient, too.

Another type of input interface could be an information message. Such text message has the same information as collected by the WWW interface, but encoded in order to fit the device and channel properties. Therefore, this type of input interface could be used by mobile devices with text input and for eMail clients. An example is given in **Tab. 3**. A voice or language selection is not necessary, because the system evaluates text for the major language used for the written text of the message and sets the selection in combination with the gender information from the selected character.

2.2. Data Creation

Using the input data provided by the user/sender the data creation step converts the given information into the desired animation. The type of animation data depends on the display or render unit at the receiver. If the receiver is able to render a 3D model in real-time, only animation parameters plus side information as well as speech data are transmitted. For movie players, the video data must be created, interleaved with the speech data, and finally encoded according to the target player properties.

DRAFT December 17, 2004 -- DRAFT December 17, 20

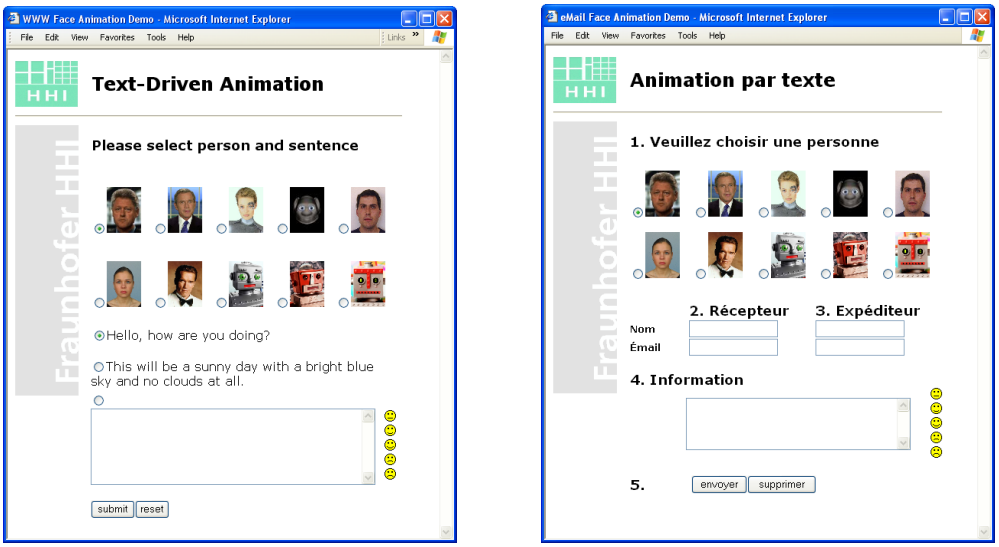

Figure 2. (left) English version of the World-Wide-Web input interface. Voice as well as sender and recipient selection are provided in a previous step of this input interface. (right) French version of the eMail or eCard input interface. Voice selection occurred at previous page.

Table 3. Information message used as input interface.

message	textual description
XXXXXXX	identification for registered users
smith@hotmail.com	contact information
+49 30 1234567	
cartoon	character, see **Tab. 2**
Hello Peter, ...	text message

2.2.1. Text-To-Speech Conversion

The initial part of the data creation step is the conversion of the given text into synthetic speech data as well as side information in the form of phonemes and phoneme duration. A phoneme is the smallest part of a spoken word and dependents on the language. Speech and lip movements are related in time. The duration of each phoneme can be measured either in time or samples of the speech data. Both types allow a precise synchronization between animation and audio data. The speech synthesis is realized by a Text-To-Speech engine. We support the AT&T Natural Voices engine, as well as BrightSpeech from Babel Technologies, and RealSpeak from ScanSoft.[13–15]

All these TTS engines come with prepared voices. Some of the TTS engines allow the adjustment of the voices for special needs. We post-process the speech data with a pitch shift mechanism. This mechanism allows us to change the main frequency of the speech, so that the voice sounds deeper or brighter adding one more degree of freedom to the system. The result is a adjusted voice, which fits more the personality of the selected character.

2.2.2. MPEG-4 Parameter Extraction

In order to create an animation, the phonemes, which are extracted from the text (see 2.2.1), are converted into MPEG-4 Facial Animation Parameters (FAP). Each FAP describes a particular facial action related to a region in the face as, e.g., chin , mouth corner, or eyes. In the 3D model, deformations are specified for all these actions, which define deviations from the neutral state. In MPEG-4, there are 66 different low level FAPs,[2] which can be superposed in order to create realistic facial animations.

DRAFT December 17, 2004 -- DRAFT December 17, 20

MPEG-4 describes the set of facial animation parameters and the range of usage. How the FAPs perform the desired facial action is not defined in the standard, but can be specified by transformation tables. Vertex movements in the 3D triangle mesh model associated to particular deformations can be taken from these tables for rendering. In order to estimate the FAPs for each frame of the video sequence from the phoneme data, the phonemes are first converted into 15 different visemes. For each viseme, measurements from real people are taken as input to estimate the corresponding set of FAPs that result in the same lip shape as given by the captured images. An analysis-by-synthesis technique is used[16] that exploits knowledge from the 3D model as well as calibration data in order to optimize the parameter set. The lip shape associated to a particular viseme is stored as a vector of 14 different FAPs. Transissions between different phonemes are interpolated by particular blending functions that allow the creation of mesh deformations for each video frame. For additional realism, random eye blinking and head motion is added to the parameter set.

2.3. Rendering of the Video

The rendering of the animation requires a frame-based deformation of the selected 3D model. The FAPs computed for each frame are applied in order to deform the mesh. Since a 3D description of the scene is available, additinal features can be applied to enhance the animation. Camera pans or zoom are only a few of the possible changes with this feature. If the animation is created on a PC based server, rendering is performed on the graphics hardware and the resulting images are saves as frames of a movie. A selection of 3D models created from a single picture is given in **Fig. 3**.

Figure 3. A set of 3D models created from single pictures.

2.3.1. Rendering on Mobile Devices

In our system, we also support rendering on mobile devices. This requires 3D graphics operations like transformations and rasterizations of a textured mesh to be implemented in software. The left sub image of **Fig. 7** shows a realization on a common Personal Digital Assistant (PDA). This device does not have a graphic chip on board, but is able to render the animation at approximately 30 fps (frames per second). The speech data are played synchroneously to the 3D animation. In order to achieve such frame rates for display devices with no graphic acceleration, modifications of the 3D model and associated texture map are required like polygon reduction, color mapping of the texture map, and 16 bit integer arithmetic. A polygon reduction of 50% as shown in **Fig. 4** increases the render rate of about 30%.

Another burst in frame rate is the conversion from floating point calculations to 16 bit integer arithmetic, since these low power mobile devices usually lack of an floating point unit and have to emulate the calculations in software. Therefore, the mesh representation and deformation calculations are done in 16 bit integer arithmetic resulting in a drastical speedup.

DRAFT December 17, 2004 -- DRAFT December 17, 20

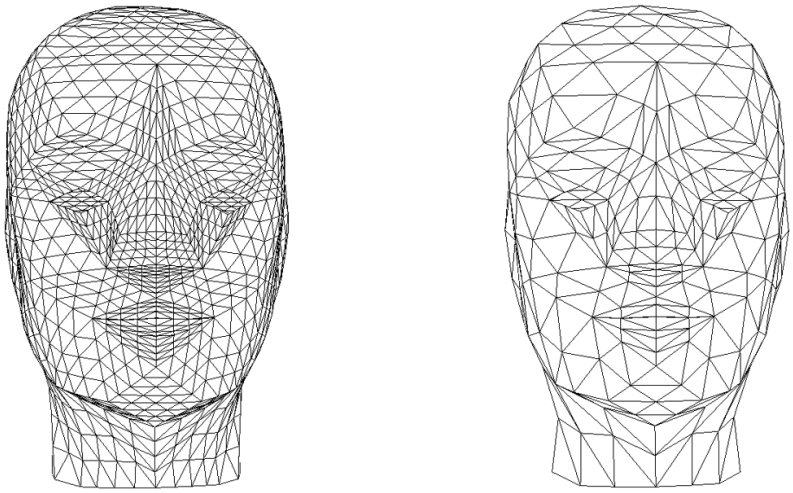

Figure 4. Polygon reduction for mobile communicator in order to increase display performance. (left) original polygon model with approximately 4000 polygons, (right) the reduced version with approximately 2200 polygons.

In order to fit texture colors with associated shading information into a 16 bit word, some color quantizations from the original 8x8x8 bit RGB texture are necessary. Since the display of the used PDA only supports a color resolution of 5x6x5 bits, only 65535 different colors can be shown. For performance reasons, the texels in the texture map only contain 3x3x3 bit of colors - the 7 remaining bits are used for internal processing. Since a color quantization with 8 steps per channel would lead to significant artifacts especially in the smoothly colored skin areas, a color map was introduced. A set of 512 colors is computed from the original texture map in order to get an optimal color quantization as shown in **Fig. 5**.

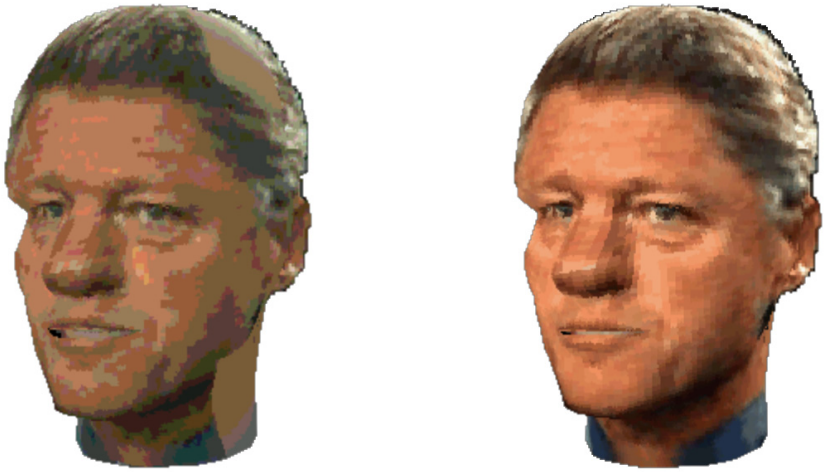

Figure 5. Color mapping for mobile devices. (left) texture map quantized to the target resolution of 3x3x3 bits, (right) optimized color mapping from 3x3x3 texture resolution to 5x6x5 display colors.

DRAFT December 17, 2004 -- DRAFT December 17, 20

2.4. Video and Audio Encoding

In the case, the rendering is performed on server side, transmission of the video is necessary. In order to create a movie, the rendered frames are encoded, interleaved with the speech data, and finally packed to the target file container. Appropriate video and audio encoder as well as file container have to be used for this step. Two different types are implemented: 3GPP container with H.263 and MPEG-4 encoders for mobile devices like mobile phones and AVI containers with MPEG-4 encoders for standard personal computer. Audio data are encoded within the 3GP container using the AMR narrow band speech encoder and MP3 for the AVI container. For Multimedia Messaging Service (MMS) messages, special constraints on the maximum file size apply. Here, automatic rate control for video and audio are used during encoding in order to fulfill the requirements.

2.5. Data Transfer

The type of data transfer depends in the same way on the render device as the data creation step. If rendering is performed on the client side, FAPs and audio information have to be transmitted to the mobile device. Because of the low amount of data for the MPEG-4 FAPs, real-time streaming can be realized by a simple data transfer protocol. We use a proprietary protocol for LAN and WLAN connections for this purpose. The data transfer protocol is shown with **Fig. 6**. It allows the transmission of animation parameters, speech data, side information and the start signal for the render engine in arbitrary order. TCP/IP is used as underlying transmission control protocol. This protocol requires a complete transfer, before rendering is started. A Real-time Transport Protocol (RTP) payload definition for phonemes and FAPs allows a continuously streaming with minimum pre-fetching of animation data.[17]

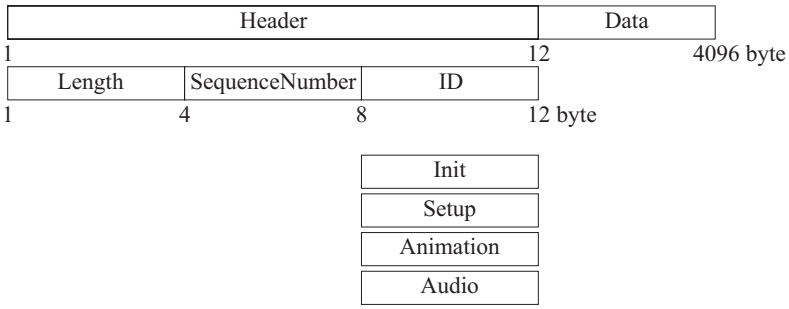

Figure 6. Data transport protocol for the transfer of animation and audio data to the mobile device. Each packet can have a length of 4096 bytes. The header with 12 bytes in equally split into three parts: length of the packet, sequence number and identification for the following data.

If the movie is rendered at the server, only one file need to be transmitted. For mobile phones, a 3GP file will be sent using the GPRS channel of the specific provider. Movie players on standard personal computer are usually connected to the World-Wide-Web by LAN or WLAN. Therefore, an HTTP-Server is used to download the animation files from the server. This system is also based on TCP/IP.

3. APPLICATION SCENARIOS

As mentioned above, the animation can be shown on many different display interfaces as depicted in **Fig. 7** enabling many new applications. In the following, we are going to describe only two of the wide range of applications using this system. First, a text messaging service, which converts SMS (Short Message Service) messages into short video clips is presented. Second, a new type of human-machine interface will be described.

3.1. SMS to MMS Conversion

This application allows the user to animate written text by adding the name of the selected character using the framework of the Short Message Service (SMS). The system analyze the written text for the major language. Together with the selected character, the defined voice will be used to create the desired animation movie. The

DRAFT December 17, 2004 -- DRAFT December 17, 20

Figure 7. Different display interfaces. (left) real-time 3D render engine implemented on a mobile communicator , (right) in car display units, (right) 3GP player on a cellular phone.

Multimedia Messaging Service (MMS) is used from this point on to transmit the video message to the receiver. The internal 3GP player on the receiver cellular phone displays the movie, which is the animated text message. An example of an animation displayed with a 3GP player is given in the right sub-image of **Fig. 7**.

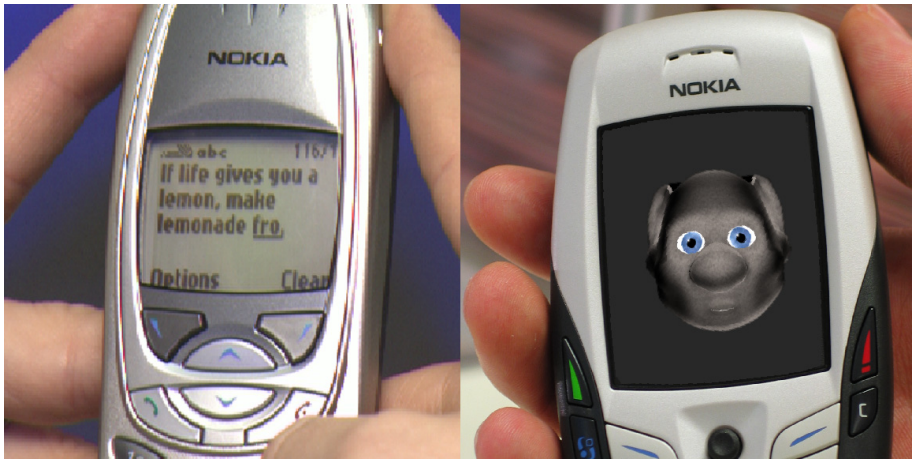

Figure 8. SMS to MMS conversion with existing infrastructure.

3.2. Human-Machine-Interface

Human-Machine interfaces usually lack personality, e.g., talking wizards, jumping paperclip. This can be circumvented by adding avatars, that help with problems and react on the user input with a voice and additional written information. The PC got a face and is not longer only a machine. Even a car can get a face for travel information, entertainment and road side assistant. News transmitted as text to the car are displayed as video with the favorite person as anchor man or woman and kids can be entertained by the latest Walt Disney character. Examples of such human-machine interfaces are given with the two middle sub-images of **Fig. 7**.

DRAFT December 17, 2004 -- DRAFT December 17, 20

4. CONCLUSION

The need of communication is essential for humans. With the fast growing connectivity (cellular phones, broadband connections to homes, etc.) new types of communications can be realized. Such new type of communication is the presented text-driven video animation. Our system is able to animate an avatar or real person from written text. With emoticons, language, gender, pitch shift, and character selections, we are able to individualize the clips which can be used for a wide range of applications. Not only the conversion from regular SMS messages to MMS animations have been implemented. Human-Machine interfaces could also profit from this technology.

ACKNOWLEDGMENTS

The work presented was developed within VISNET, a European Network of Excellence (http://www.visnet-noe.org), funded under the European Commission IST FP6 programme.

The authors would like to thank Jens Güther for providing fruitful ideas, software as well as hardware knowledge for this work.

REFERENCES

1. T. Wiegand, G. B. G. Sullivan, and A. Luthra, "Overview of the H.264/AVC Video Coding Standard," *IEEE Transactions on Circuits and Systems for Video Technology* , July 2003.
2. ISO/IEC International Standard 14496-2, *Generic Coding of Audio-Visual Objects – Part2: Visual*, 1998.
3. P. Eisert, "MPEG-4 Facial Animation in Video Analysis and Synthesis," *Journal of Imaging Systems and Technology* **13**, pp. 245–256, March 2003.
4. J. Ostermann, "Face Animation in MPEG-4," in *MPEG-4 Facial Animation - The Standard Implementation and Applications*, I. S. Pandzic and R. Forchheimer, eds., Wiley, 2002.
5. T. Dutoit, "High-Quality Text-to-Speech Synthesis : an Overview," *Journal of Electrical & Electronics Engineering, Australia: Special Issue on Speech Recognition and Synthesis* **17**(1), pp. 25–37, 1997.
6. S. Kshirsagar, M. Escher, G. Sannier, and N. Magnenat-Thalmann, "Multimodal Animation System Based on the MPEG-4 Standard," in *Proceedings of the International Conference on Multimedia Modeling (MMM 99)*, pp. 215–232, (Ottawa, Canada), October 1991.
7. J. Ostermann, M. Beutnagel, A. Fischer, and Y. Wang, "Integration of Talking Heads and Text-to-Speech Synthesizers for Visual TTS," in *Proceedings of the International Conference on Speech and Language Processing (ICSLP 98)*, (Sydney, Australia), December 1998.
8. S. Kshirsagar, C. Joslin, W. Lee, and N. Magnenat-Thalmann, "Personalized Face and Speech Communication over the Internet," *IEEE Signal Processing Magazine* **18**, pp. 17–25, May 2001.
9. J. Ostermann, "E-Cogent: An Electronic Convincing aGENT," in *MPEG-4 Facial Animation - The Standard Implementation and Applications*, I. S. Pandzic and R. Forchheimer, eds., Wiley, 2002.
10. S. Beard, J. Stallo, and D. Reid, "Usable TTS for Internet Speech on Demand," in *Proceedings of the Talking Head Technology Workshop (OZCHI)*, (Perth, Australia), November 2001.
11. Third Generation Partnership Project (3GPP), *TS 26.140 v6.0.0; Multimedia Messaging Service (MMS); Media formats and codecs (Release 6)*, September 2004.
12. Third Generation Partnership Project 2 (3GPP2), *C.S0050-0; Version 1.0; 3GPP2 File Formats for Multimedia Services*, December 2002.
13. AT&T, *Natural Voices^{TM}; Text-To-Speech Engines (Release 1.4)*, 2002.
14. Babel Technologies home page, *http://www.babeltech.com*.
15. ScanSoft home page, *http://www.scansoft.com/realspeak/demo/*.
16. P. Eisert and B. Girod, "Analyzing facial expressions for virtual conferencing," *IEEE Computer Graphics & Applications* **18**, pp. 70–78, September 1998.
17. J. Ostermann, J. Rurainsky, and R. Civanlar, "Real-time streaming for the animation of talking faces in multiuser environments," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, (Scottsdale, Arizona, USA), May 2002.

DRAFT December 17, 2004 -- DRAFT December 17, 20