Lifelike Talking Faces for Interactive Services

ERIC COSATTO, MEMBER, IEEE, JÖRN OSTERMANN, SENIOR MEMBER, IEEE,
HANS PETER GRAF, FELLOW, IEEE, AND JUERGEN SCHROETER, FELLOW, IEEE

Invited Paper

Lifelike talking faces for interactive services are an exciting new modality for man–machine interactions. Recent developments in speech synthesis and computer animation enable the real-time synthesis of faces that look and behave like real people, opening opportunities to make interactions with computers more like face-to-face conversations. This paper focuses on the technologies for creating lifelike talking heads, illustrating the two main approaches: model-based animations and sample-based animations. The traditional model-based approach uses three-dimensional wire-frame models, which can be animated from high-level parameters such as muscle actions, lip postures, and facial expressions. The sample-based approach, on the other hand, concatenates segments of recorded videos, instead of trying to model the dynamics of the animations in detail. Recent advances in image analysis enable the creation of large databases of mouth and eye images, suited for sample-based animations. The sample-based approach tends to generate more naturally looking animations at the expense of a larger size and less flexibility than the model-based animations.

Beside lip articulation, a talking head must show appropriate head movements, in order to appear natural. We illustrate how such "visual prosody" is analyzed and added to the animations. Finally, we present four applications where the use of face animation in interactive services results in engaging user interfaces and an increased level of trust between user and machine. Using an RTP-based protocol, face animation can be driven with only 800 bits/s in addition to the rate for transmitting audio.

Keywords**—***Avatar, computer graphics, face animation, MPEG-4, sample-based graphics, speech synthesizer, text-to-speech (TTS), video-based rendering, visual text-to-speech (VTTS).*

I. INTRODUCTION

At the start of the new millennium, telecommunication is slated to fully embrace "data" over Internet Protocol (IP) networks in the form of multiple media such as voice, video, documents, database accesses, etc. More and more devices, from telephones to Personal digital assistants (PDAs) and PCs, will enable communications over IP networks in multiple modalities, including "video" in addition to the traditional "voice" communication. Increasingly, human-to-human communication will be amended by communication between humans and machines for such applications as e-commerce, customer care, and information delivery in services [1].

Today, human–machine communication is dominated by the input of typed text plus mouse clicks, while the machine produces text and graphics output. With recent advances in speech recognition, natural language interpretation and speech synthesis, however, conversational interfaces are finding wider acceptance. The next step in giving human–machine interfaces the look and feel of human–human interaction is the addition of visual elements. Image analysis enables the recognition of faces, or of other objects for a visual input, while animation techniques can synthesize human faces providing spoken output by a machine. In this paper, we focus on this latter aspect, namely, on how to synthesize human faces, with an emphasis on techniques that produce video-realistic animations. The goal is to provide computers with synthesized faces that look, talk, and behave like real, human faces.

Generating lifelike animated faces remains a challenging task despite decades of research in computer animation. To be considered natural, a face has to be not just photo-realistic in appearance, but must also exhibit proper postures of the lips, synchronized perfectly with the speech. Moreover, realistic head movements and emotional expressions must accompany the speech. We are trained since birth to recognize faces, and to scrutinize facial expressions. Consequently, we are highly sensitive to the slightest imperfections in a synthesized facial animation. Only very recently, technology has advanced to a point where talking heads can be synthesized with a quality comparable to recorded videos. This is due to progress in text-to-speech (TTS) synthesis as well as in computer graphics. It is noteworthy that in both areas, major advances in naturalness have been achieved through the application of sample-based techniques (e.g., [2], [3], Section IV). We call talking faces that are driven by a speech synthesizer visual TTS (VTTS).

Synthesized faces are an integral part of animated movies and video games, but beyond entertainment, they are slow in

Manuscript received October 31, 2002; revised March 12, 2003.

The authors are with the AT&T Labs–Research, Middletown, NJ 07748 USA (e-mail: eric.cosatto@ieee.org; joern.ostermann@ieee.org; hpgraf@ieee.org; j.schroeter@ieee.org).

Digital Object Identifier 10.1109/JPROC.2003.817141

finding widespread use. Despite a wealth of data suggesting their potential benefits, only a few talking heads appear in such commercial applications as customer service. The animated face is associated by most people with entertainment, and, until recently, the quality of VTTS was not sufficient to have synthetic faces act as stand-ins for real humans. This fact had many people question the economic viability of talking heads in "serious" applications. In the world of down-to-earth business economics, the introduction of new technologies, such as VTTS, into business and consumer services has proven quite difficult. The chicken-and-egg problem of only introducing new technologies after they have received favorable market reception makes it necessary to carefully plan a multistep strategy that involves usability studies, proof-of-concept studies, and careful choice of initial test markets. This paper tries to sort out some of these issues by addressing technical problems and solutions, by discussing how to evaluate VTTS, and by describing some potential applications.

The paper is organized as follows. The remainder of the introduction provides a more detailed overview of talking head systems in computer user interfaces (Section I-A) and their implementations (Section I-B). Section II discusses some of the system issues that enable effective talking head interfaces. Sections III and IV focus on the technology of generating talking heads using conventional three–dimensional (3-D) computer graphics and sample-based techniques, respectively. Section V analyzes the question of *visual prosody*, the movements and behavioral patterns that accompany speech, and Section VI addresses the problem of evaluating the quality of synthesized speech. In Section VII, several applications are described, and some concluding remarks round out the paper in Section VIII.

A. Talking Heads in Computer User Interfaces

Animated faces have many potential applications, for example, in e-learning, customer relations management, as virtual secretary, or as your representative in virtual meeting rooms. Many of these applications promise to be more effective if the talking heads are video-realistic, looking like real humans. When buying something on a Web site, a user might not want to be addressed by a cartoon character. However, streaming or live video of a real person is in most cases not feasible because the production and delivery costs are far too high. Similar arguments apply to e-learning applications. Several researchers have found that a face added to a learning task can increase the attention span of the students [4]–[6]. Yet producing videos is prohibitively expensive for most e-learning tasks. If an application is accessed over the Internet, there is the additional difficulty of a limited bandwidth that often prevents streaming or live videos. With synthetic faces, it is possible to achieve a far higher compression than usual with compressed videos; hence, they can be presented over narrowband modem connections.

One important application of animated characters has been to make the interface more compelling and easier to use. For example, animated characters have been used in presentation systems to help attract the user's focus of attention, to guide the user through steps of a presentation, as well as to add expressive power by presenting nonverbal conversational and emotional signals [7], [8]. Animated guides or assistants have also been used with some success in user help systems [9], [10], and for user assistance in Web navigation [11]. Personal character animations have also been inserted into documents to provide additional information to readers [12].

Recently, character animation has started to play a significant role in interface design of communication or collaboration systems. There are several multiuser systems that currently use avatars, which are animated representations of individual users [13], [14]. However, in many cases, the avatar authoring tools and online controls remain cumbersome. Furthermore, the cues that are needed to mediate social interaction in these new virtual worlds have been slow to develop, and have resulted in frequent communication misunderstandings [15]. Nevertheless, the enormous popularity of Internet chat applications suggests considerable future use of avatars in social communication applications.

The use of facial animation has been the primary research focus of several studies of multimodal interfaces. One important area of inquiry has been the nature of the social interaction in applications that use facial animation. In one interview task, for example, researchers found that users revealed more information, spent more time responding, and made fewer mistakes when interacting with an animated facial display compared with a traditional paper and pencil questionnaire [16]. Furthermore, the increased responsive effects were greater when a *stern* facial animation was used compared to a more neutral face. In a second study, subjective reports about an interview with an animated face revealed that users attributed personality traits to the face, reported that they were more alert, and presented themselves in better light compared with an interview using only text [17]. Positive personality attributes such as self-confidence, friendliness, and cheerfulness were rated higher when the animated character showed emotional responses like joy and sadness [18]. Finally, users exhibited a higher degree of cooperation when interacting with animated partners in a social dilemma game. The presence of an animated cartoon increased the degree of cooperation or trust by 30% [19]. The fact that the animated face was *human* was important, as the same cooperative interaction was not observed using animated faces of dogs [20]. The studies on the effects of facial animation on human computer interactions were all based on face animation systems displaying obviously synthetic faces. We believe that the positive influence of animated characters will increase further, if we are able to produce lifelike talking faces that are indistinguishable from a real human.

B. Talking Head Implementations

Two basic approaches to modeling and synthesizing talking heads exist: the *model-based* approach, which focuses on shape, and the *sample-based* approach, which focuses on texture. While shape information allows flexible rendering from any viewpoint, texture provides for realistic appearance. The tradeoff between shape and texture is illustrated in Fig. 1.

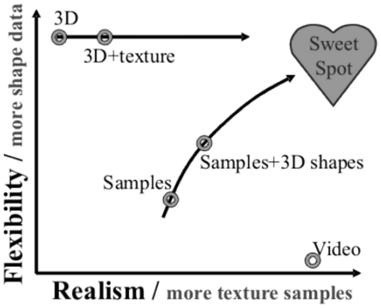


Fig. 1. Flexibility versus realism for different approaches. While recorded video offers the most realism, it lacks flexibility: what has been recorded is what you can show. On the other hand, a fully model-based 3-D talking head can be shown from any direction, under different lighting conditions and can produce any deformations, but, because of simplifications, it lacks realism. The sample-based approach trades off some flexibility (the head can only be shown under a reduced range of views) for the realism provided by sampled textures. As 3-D information is combined with the sample textures, more flexibility is given to the model, bringing such systems closer to the "sweet spot."

The traditional *model-based* technique starts with a mesh of 3-D polygons that define the shape of the head and its various parts in high detail. An "intelligent" layer is then added to allow the mesh to be parametrically deformed to perform facial actions. Finally, a texture image is mapped over the mesh to render the fine details of the skin and facial parts. Current interest for this technology is clearly shown by its inclusion in the MPEG-4 standard (see Section III).

The model-based approach has the advantage of a small footprint. The model requires only a single texture image and the number of polygons in the 3-D mesh can be reduced, making it possible to produce head models of only a few kilobytes in size. Many commercial products, such as those from Anthropics [21], face2face [22], Famous3D [23], LifeFX [24], Lipsinc [25], and Pulse3D [26], feature very small footprint players as well as very small file size models, making them well suited for low-bandwidth situations such as dial-up Internet connections. In most of these systems, the model can be created from a single photograph, simplifying the task of model creation. While attractive in their simplicity and download speed, these systems produce animations that lack natural lip articulation, expressions, and head movements. Large amounts of development work has gone into these systems to make them easy and quick to download, install, and integrate into such programs as e-mail clients. Indeed, many of the hurdles in having talking heads technology widely accepted eventually reside in solving these issues.

In contrast to companies targeting real-time animation of talking faces on conventional PCs, companies working in movie production and many researchers are pursuing a more sophisticated model-based approach, attempting to model the head in all its details. For example, in [27], the skin is modeled with several layers of spring lattices (to account for the different elasticity of skin and fatty tissues), and muscle actions are applied that simulate wrinkling. Kalra *et al.* [28] perform facial animations using free-form deformation to adapt volumetric objects by manipulating control points arranged in 3-D cubic lattices. For comprehensive surveys of these techniques, see [29] and [30]. However, in a departure from purely analytical modeling, many of these systems use a captured image of the face that is texture mapped onto a polygon mesh. This shortcut avoids the complication of modeling the surface of the skin (pores, facial hair, dimples, oiliness, etc.). A tradeoff of this shortcut, however, is that this single-texture image is constantly stretched to accommodate skin movements and deformations such as jaw rotations when the mouth opens. Such image stretching substantially alters the visual quality of the skin texture and results in an unnatural appearance. These artifacts are perceived as awkward by most viewers and tend to be strongly disliked. In order to avoid these artifacts for movie productions, surface properties are modeled using transparency and reflectance functions. However, these systems will not be real-time capable in the foreseeable future.

An additional difficulty in the analytical model-based approach is to model the behavior of the mouth as it articulates speech. The complex dynamics of speech articulation are often approximated using coarticulation models [31]. These models define key mouth shapes (typically acquired through extensive observations of how people articulate) as well as a functional model to interpolate between these key frames to produce transition mouth shapes. Section II covers coarticulation models in more details.

The second technique is often referred to as *sample-based*, image-based, or data-driven, and focuses on texture. Instead of modeling facial movements by deforming a 3-D mesh, it uses recorded video images of facial parts to render these movements directly at the image level. It bypasses the complexities of analytical modeling by learning the model from recorded samples of data (usually video and audio of a person articulating speech and producing facial expressions). Hence, both the appearance of the mouth as it articulates and its complex temporal behavior in time are captured.

Two directions exist within the sample-based approach. One is to record a limited set (typically less than 100) of key image samples (often called visemes) and synthesize all transitions between these key images using image processing techniques. Ezzat *et al.* [32], [33] have demonstrated a sample-based talking head system that uses morphing to generate intermediate appearances of mouth shapes from a set of recorded mouth samples. The dynamics of speech articulation is captured through a statistical model of the phonemes (modeled by Gaussians) in a parameter space. This system does not yet address the synthesis of head movements and facial expressions.

The other direction consists of recording many samples of facial parts so that all their appearances are captured. A talking head synthesis technique based on recorded samples that are selected automatically has been proposed in [34]. This system uses video snippets of triphones (three subsequent phonemes) as samples. Since these video snippets are parameterized with phonetic information, the resulting database is very large. Moreover, this parameterization can only be applied to the mouth area, precluding the use of other facial parts such as eyes and eyebrows that are carrying important conversational cues. In Cosatto *et al.* [35], [36] variable length units of video are combined by finding an optimal path in a dynamically built animation graph. Details of this system are discussed in Section IV. These systems implicitly model coarticulation by capturing entire video segments of articulation. Furthermore, they also capture the specific ways a person articulates.

A recent approach that promises to combine realism and flexibility consists in sampling simultaneously the texture and the 3-D geometry of faces as they articulate speech or produce facial expressions. The availability of low-cost 3-D scanners such as the ones from Cyberware [37] or Eyetronics [38] is bringing this approach closer to reality. However, many difficulties remain and in particular, capturing high-quality textures is difficult and sometimes impossible to do simultaneously with recording the 3-D shape. For example, in the work of Guenter *et al.* [39], hundreds of fiducial dots are glued on the talent's head and the images are captured by a setup of six cameras. Resulting texture images undergo processing to remove the dots and a second processing step to combine the six separate views. In the work of Pighin *et al.* [40], a setup phase consisting of manual labeling of 100 points on the face is necessary to register the 3-D shape of the head from a set of photographs, making this approach unsuitable for speech synthesis, where hundreds of images would have to be marked. Another interesting research direction consists in finding principal directions of variance in faces' shapes using a principal components analysis (PCA) on 3-D range data. Blanz *et al.* [41] have demonstrated a system producing transitions between heads of different people, as well as caricatures. Recent work by Kalberer *et al.* [42] is aimed at producing speech animations, using similar techniques.

II. System Overview

We first describe the components and interfaces of a basic face animation system. Then we discuss how to drive talking faces using text or recorded speech. Section II-D concludes with an overview of face models and their respective data sets.

A. Elements of a Face Animation System

We consider that the output of a face animation system is the rendering a face model onto a display for presentation to the user. We also consider four types of input to such a system:

1) ***Audio:*** The voice of the face model is played back through the loudspeakers of the user's terminal. This audio signal might be a recorded voice or synthetic speech computed by a TTS synthesizer.
2) ***Phonemes*** and related timing information: Phonemes (e.g., /b/, /j/, /o/) are the speech sounds that make up spoken words. The timing information specifies the start time and duration of a phoneme in the audio signal. Phonemes are used to determine the appropriate mouth shape for the face model.
3) ***Face animation parameters* (FAPs):** These parameters directly affect the face model. Usually, they are used to define nonspeech-related motions of the face. In this case, these parameters include head turns and nods, gaze direction, as well as emotions like joy or surprise.
4) ***Text:*** The text spoken in the audio signal may be analyzed in order to compute the appropriate visual prosody. Visual prosody encompasses skull movements (nods, et.) as well as facial movements (eyebrow rise, eye blinks, etc.) that accompany speech. Most people are not aware of their visual prosody. However, the lack thereof immediately makes a face look unengaged and/or unnatural.

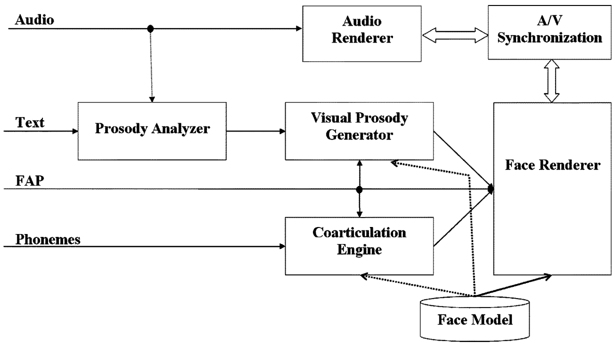


Fig. 2. Basic architecture for animating a talking face.

The look of a face model in its neutral state is defined by the shape and texture of the face model. The dynamics of a face model are defined by the way it reacts to the input. For example, each face model should have its own smile, where the smile is not only defined by its look and amplitude but also by the temporal change of the amplitude. Similar arguments can be made for almost any movement of a face. The personality and mood of a face model might also influence the dynamic behavior of a face [43].

Fig. 2 shows the high-level architecture for animating talking faces. The *face renderer* and the *audio renderer* compute the audiovisual output on the client terminal. An *audiovisual synchronization* module controls the temporal alignment of the audio and video presentation. The input signals of the audio and video renderer are either computed locally or received over a network connection. The audio signal carries the voice of the talking face. An audio compression algorithm may be used to increase bandwidth efficiency. The face renderer takes as input FAPs that describe how to change the face model from its neutral state. There are three sources for FAPs: they may be directly provided as an input signal, they may be derived from the input text, or they may be generated from the phonemes that are read by the face animation system.

Coarticulation: The *coarticulation engine* computes the lip shapes of the face model depending on the specific sequence of phonemes and their timing. A simple mapping between phonemes and lip shapes does not exist. The same lip movements are involved in producing different speech sounds. A viseme, as defined in [44], represents phonemes that contain optically similar sequences of oral-cavity movements. A viseme is an experimental concept [45], and different mappings of phonemes to visemes exist. Often, visemes are just defined as lip shapes [46].

Humans move lips, tongue, and jaw quite rapidly while talking. Coarticulation occurs because the different speech production processes, and the different articulators involved, combine with one another with different timing patterns. Coarticulation may be generally defined as "the overlapping of adjacent articulations" [47]. All languages exhibit coarticulatory phenomena, though in varying ways. One of the ways in which they differ is in directionality. Compare English /k/ in "peak" /pik/ and "keep" /kip/, "caw" /k/ and "hawk" /hk/, then in "peak" and "hawk." You should find that /k/ undergoes greater influence (alteration of place of articulation) from the following vowel than from a preceding vowel. This direction of influence is called anticipatory or regressive coarticulation. In other languages, such as French and Italian, a preceding vowel will exert greater influence—perseverative or progressive coarticulation [47]. Therefore, the implementation of a coarticulation engine is language dependent. Furthermore, coarticulation differs between people—although these differences may not be as significant as those between languages.

Several groups developed model-based coarticulation engines [31], [45], [48]. These engines are based on measurements of lip and mouth movements of a person speaking in a certain language. Therefore, they model the coarticulation of a specific person. Based on the phonemes and their duration, these models compute, for each time instant, an appropriate mouth shape. These models allow presenting talking faces where the lip movement appears to be synchronized with the spoken words. However, a shortcoming of these coarticulation models is that they produce identical coarticulation and mouth movements for different face models. In case we use a face model representing a particular human, we will, in general, not be able to recreate the mouth movements of this human. Sample-based coarticulation overcomes this shortcoming by providing a face model with its own coarticulation model (see Fig. 2) implemented as a database (Section IV).

Visual Prosody: When humans speak, they use *prosody* (intonation, word-level stress, etc.) to indicate syntax, turn taking in conversational interactions, types of utterance such as questions and statements, and the acoustic correlates of people's attitudes and feelings. The elements of prosody are derived from the acoustic characteristics of speech. They include the pitch or fundamental frequency, the length or duration of individual phonemes, and their loudness or intensity. All these forms are present in varying quantities in every spoken utterance [49], [50].

When we articulate speech, not only the lips, tongue, and jaw are moving, but typically, also the whole head moves in synchrony with the speech and its prosody. Much information related to phrasing, stress, intonation, and emotion are expressed by, for example, nodding of the head, raising and shaping of the eyebrows, as well as eye movements and blinks [48]. Since many of these movements are tied so closely to the prosody of speech, we call them *visual prosody*.

Visual prosody should also be taken into account in a face animation system, not only because it may carry important nonverbal information, but also because visual prosody makes the face look alive. It is the combination of speech articulation with all other facial movements that gives us the impression of natural speech. These movements are more difficult to model in a general way than the articulatory movements, since they are optional, and highly dependent on the speaker's personality, mood, purpose of the utterance, etc. However, speakers commonly raise their eyebrows at the end of a question or at a stressed syllable. Similarly, eye blinks tend to occur at the beginning of a word.

Little quantitative information exists in the literature about prosodic facial movements. Prosodic nods are mentioned sometimes in connection with animating faces, e.g., in [29], but few details are given. Eckman and Friesen studied extensively emotional expressions of faces [51]. They also describe nonemotional facial movements that mark syntactic elements of sentences in particular endings. Bavelas and Chovil classify facial expressions into semantic and syntactic categories and provide quantitative information about the frequency of the different types [52]. Gaze in dialogue situations is studied in [53], [54]. In [55] a statistical model is developed connecting acoustic features with FAPs. Such studies are valuable in guiding us where to place facial expressions. On the other hand, in the psychology literature, facial expressions are typically evaluated by human observers, and little quantitative information is provided about speed and amplitude of head rotations or of the movements of facial features. The correlation between prosodic events and head movements is studied quantitatively in [56] and in Section V later.

In order to provide visual prosody, a *prosodic analyzer* evaluates the spoken text (see Fig. 2). The analysis may also consider the audio signal. Using the prosodic information of the text, the *Visual Prosody Generator* computes the appropriate FAPs. While a few general rules apply to most speakers, visual prosody varies from person to person. Therefore, the face model may provide control parameters or a database for the prosody generator (Section V). Several face animation systems use rule-based visual prosody [19], [57], [58].

While prosody is mainly controlled by the syntax of speech, emotions are dependent on the state of mind of the speaker as well as the semantics of the speech. The emotional display of an animated face can be controlled by sending the appropriate FAPs to the face animation system.

B. Text-Driven Face Animation

If we want to drive talking faces as part of an automatic and intelligent dialogue system, we have to create speech from the text that the dialogue system creates in response to the user input. The speech is usually rendered by a TTS system. TTS systems synthesize text based on the phonemes that correspond to the words in the text. Therefore, any TTS can be used to drive a face animation system [31], [57], [59], provided the phoneme timing information is accessible.

The TTS system analyzes the text and computes the correct list of phonemes, their duration, appropriate stress levels, and other parameters. Modern TTS systems create prosodic information in order to enable natural sounding speech synthesis. The text and related phonemes and their durations are sent to the face animation system. Alternatively, the prosodic information of the TTS engine may be sent to the face animation system and its prosody analysis may be skipped. Finally, the TTS engine computes the audio signal. Some systems allow for bookmarks in the text that animate nonspeech FAPs like emotions or head turns synchronized with the text. In this case, the TTS determines the start time of the FAP as it is determined by the position of the bookmark in the text. A very simple example is "I start smiling :-) now" [60], [61].

While early concatenative synthesizers (i.e., synthesizers that stitch together snippets of speech to generate an output utterance [62]) used very few prototypical units for each class of inventory elements mainly due to limitations in computational resources, new speech synthesizers that leverage the recent increases in computation and storage capabilities synthesize a voice that is almost indistinguishable from human recordings [63]. While we are used to synthetic faces with natural-sounding voices (cartoons), the synthetic sounding voices of earlier TTS made face animation with TTS unattractive. With the latest TTS technology, however, we are now able to automatically create convincing animations of cartoons, and with the latest face animation technology, we can even create animations that are indistinguishable from recorded video (see Section VI-B).

C. Voice-Driven Face Animation

Despite (almost) natural-sounding TTS, human speakers can create more expressive speech due to their capability to adapt intonation, emphasis, and pauses easily to the semantics of the spoken word [64]. Therefore, many researchers investigate how to animate a talking face from a natural voice. There are two different approaches.

Lip Shapes from Audio: In order to derive lip shapes directly from audio, neural nets [65], hidden Markov models (HMMs) [66], [67], linear prediction analysis [68], and sound segmentation [69] were developed. Their output—FAPs representing the lip shapes—can be directly fed into the face renderer (see Fig. 2). These systems are real-time capable. They are usually good in detecting mouth closures, and, thus, they provide decent lip synchronization. However, it is obvious to the viewer that the lip movements are not derived from a naturally speaking person.

Phonemes from Audio: Speech recognition techniques are able to recognize the words in recorded speech. The text can then be used to align the phonemes of text and the audio signal. In such a way, we are able to hand text as well as phonemes with their durations to the face animation system (see Fig. 2). If real-time performance is not required, the recorded speech can be transcribed manually, thus avoiding recognition mistakes. Then the text is aligned with the audio. In the case of high-quality recordings, the automatic alignment procedures [64], [70] work very well, resulting in high-quality mouth animations comparable to those achieved using a TTS engine. Sample-based face animation (Section IV) with recorded audio can look so natural that it is indistinguishable from recorded video for most viewers (see Section VI).

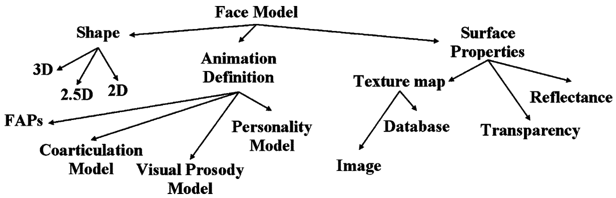


Fig. 3. A face model may contain several sets of data.

Performance-driven face animation extracts FAPs from motion capture or video data and uses these to animate a face [71]. The audio signal is captured from the performer and played back; hence, the audio and visual data of the face animation system are processed separately. We do not consider these techniques here.

D. Face Model Overview

The purpose of a face model is to describe the face such that it can be animated. Depending on the required realism, a face model may consist of two or more parameters sets. Typically, the shape of the face model is described as a mesh using two-dimensional (2-D) or 3-D coordinates (see Fig. 3). Some face animation systems use a combination of 2-D planes in order to create a coarse approximation of a 3-D shape with surface texture, thus providing the illusion of a complete 3-D shape [35]. These models are sometimes referred to as 2.5-D. A 3-D face model enables simple content creation. The face model can be integrated into any environment without enforcing any limitation on camera movements. Applications in human–computer interfaces usually require the face model to look at the user, so that 2-D and 2.5-D face models can be used.

The face renderer needs to know how to change the face model when applying FAPs. This knowledge is captured in animation definition parameters. Very simple face animation systems use an animation definition built into the face renderer. Thus, they are only able to animate face models with one particular topology. This shortcoming becomes obvious if several face models are animated simultaneously and they all show the same facial expression. MPEG-4 face animation as well as other systems use face models that include the definition of FAPs such that every model can have its own implementation of a FAP, for example, for a smile [72], [73]. Some face models will also contain a coarticulation model ([33] and Section IV). This enables creating a face model with a unique way of visual articulation. A visual prosody model helps to personalize the nonspeech-related animation while the model speaks. Some advanced face models do incorporate a personality model that controls the behavior of the face model during a dialogue [43], [58].

Several parameters may be used to describe the surface properties of a model. They include the transparency of an object that defines the opacity or translucency. The surface reflectance defines what and how much the surface reflects of the incident light. That is, is the surface glossy or matte? What color does it have? What are the patterns on the surface? The reflectance is influenced by the microscopic roughness of the surface and the surface material. The roughness is not captured in the shape that describes macroscopic properties. A complete definition of the surface reflectance allows for placing the face model into a scene where it will look correctly illuminated. It will reflect onto other objects and show reflections of the environment on its surface. Achieving this realism requires extensive modeling efforts in defining the reflectance as well as compute-intensive ray tracing algorithms for rendering the face. These techniques are, therefore, not suitable for real-time application but are common in movie productions [74]. Again, this flexibility might not be required in many human computer interface applications. A texture map may be used to simplify defining surface properties [75]. In its simplest form, the texture map is an image glued onto the shape of the face model. Texture maps are the dominant method of describing surface properties for face animation in interactive systems. Using a database of images increases realism as it avoids stretching texture maps during face model deformation (Section IV).

III. Three-Dimensional Faces and MPEG-4

Since computer face animation became popular, many groups started to create face animation systems using 3-D face models. In this section, we describe how MPEG-4 implements faces. MPEG-4 is an object-based multimedia compression standard, which allows for encoding of different audiovisual objects (AVO) in the same scene independently. The visual objects may have natural or synthetic content, including arbitrary shape *video objects*, special synthetic objects such as a human face and body, and generic 2-D/3-D objects like indexed face sets, which define an object surface by means of vertices and surface patches.

The representation of synthetic visual objects in MPEG-4 is based on the prior Virtual Reality Modeling Language (VRML) standard [76]–[78] using nodes such as *Transform*, which defines rotation, scale, or translation of an object, and *IndexedFaceSet*, describing the 3-D shape of an object by an indexed face set. However, MPEG-4 is the first international standard that specifies a compressed binary representation of animated synthetic audiovisual objects and scenes. The encoders do enjoy a large degree of freedom in how to generate MPEG-4 compliant bit streams. As specified in MPEG-4 systems, decoded audiovisual objects can be composed into 2-D and 3-D scenes using the Binary Format for Scenes (BIFS) [76], which also allows for implementation of generic animation of objects and their properties. Very efficient special purpose animation techniques are defined for face and body animation in MPEG-4 Visual [46]. MPEG-4 also provides structured audio tools for sound synthesis and a TTS interface (TTSI) for speech synthesis [79]. Body animation and 3-D mesh compression are supported in MPEG-4 to complement face animation; however, they are not covered here [46], [76], [80], [81].

A. Face Animation Parameters

MPEG-4 specifies a set of FAPs, each corresponding to a particular facial action deforming a face model in its neutral state. The FAPs are based on the study of minimal perceivable actions and are closely related to muscle action [28], [82], [83]. Sixty-six FAPs define low-level actions like head and eye rotation or motion of face deformation by moving a feature point on the face (see Fig. 4). In addition, 14 visemes and six facial expressions (joy, sadness, surprise, disgust, fear, and anger) are specified. (see Fig. 5). The FAP value for a particular FAP indicates the magnitude of the corresponding action, e.g., a big versus a small smile. A particular facial action sequence is generated by deforming the face model from its neutral state according to the specified FAP values for the corresponding time instant before being rendered onto the screen.

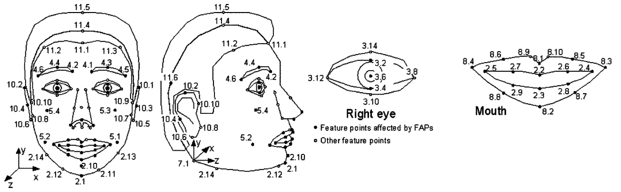


Fig. 4. Subset of feature points defined by MPEG-4. FAPs are defined by motion of feature points.

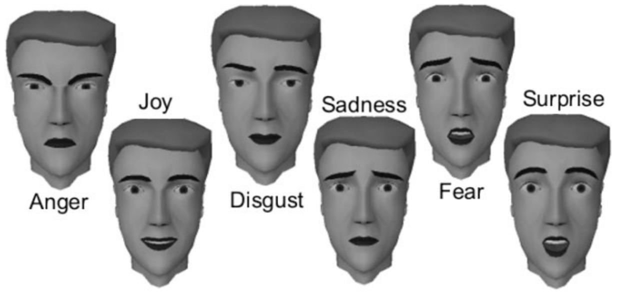


Fig. 5. Primary facial expressions.

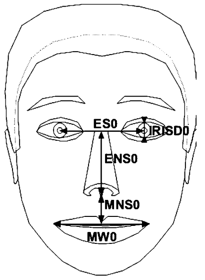


Fig. 6. FAPUs (from [46]).

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). FAPUs are defined as fractions of distances between key facial features (see Fig. 6). These features allow interpretation of the FAPs on any facial model in a consistent way, producing reasonable results in terms of expression and speech pronunciation.

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 for each FAP the animation rules for the model.

B. Face Model Specification

MPEG-4 allows the encoder to specify completely 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 using FaceDefTables that specify how this model is deformed by the FAPs [84].

1) Static Geometry Using a Scene Graph: The static geometry of the head model is defined with a scene graph specified using MPEG-4 BIFS [76]. For the purpose of defining a head model, BIFS provides the same nodes as VRML. VRML and BIFS describe geometrical scenes with objects as a collection of nodes, arranged in a scene graph. Three types of nodes are of particular interest for the definition of a static head model. A *Group* node is a container for collecting child objects: it allows for building hierarchical models. For objects to move together as a group, they need to be in the same *Transform* group. The *Transform* node defines geometric affine 3-D transformations like scaling, rotation, and translation that are performed on its children. When *Transform* nodes contain other *Transforms*, their transformation settings have a cumulative effect. Nested *Transform* nodes can be used to build a transformation hierarchy. An *IndexedFaceSet* node defines the geometry (3-D mesh) and surface attributes (color, texture) of a polygonal object. Texture maps are coded with the wavelet coder of the MPEG texture coder. Since the face model is specified with a scene graph, this face model can be easily extended to a head and shoulder model.

2) Animation Rules Using FaceDefTables: A FaceDefTable defines how a model is deformed as a function of the amplitude of the FAP. It specifies, for a FAP, which Transform nodes and which vertices of an IndexedFaceSet node are animated by it and how. FaceDefTables are considered part of the face model.

Animation Definition for a Transform Node: If a FAP causes solely a transformation like rotation, translation, or scale, a Transform node can describe this animation. The FaceDefTable specifies the type of transformation and a neutral value for the chosen transformation. During animation, the received value for the FAP and the neutral value determine the actual value.

Animation Definition for an IndexedFaceSet Node: If a FAP, such as a smile, causes flexible deformations of the face model, the animation results in updating vertex positions of the affected IndexedFaceSet nodes. The affected vertices move along piecewise linear trajectories that approximate flexible deformations of a face. A vertex moves along this trajectory as the amplitude of the FAP varies. The FaceDefTable defines for each affected vertex its own piecewise linear trajectory by specifying intervals of the FAP amplitude and 3-D displacements for each interval. Fig. 7 shows two phases of a left eye blink (plus the neutral phase) that have been generated using a simple animation architecture [84].

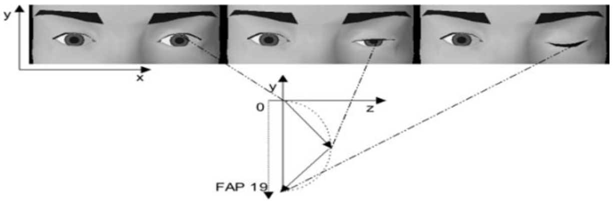


Fig. 7. Neutral state of the left eye (left) and two deformed animation phases for the eye blink (FAP 19). The FAP definition defines the motion of the eyelid in negative y-direction, the FaceDefTable defines the motion of the vertices of the eyelid in x and z direction. For FAP 19, positive FAP values move the vertices downwards ([72]).

IV. Sample-Based Talking Faces

In this section, we describe the sample-based approach to facial animation, developed to generate video-realistic synthetic talking heads. First, we record a person uttering a corpus of phonetically balanced sentences. Then the video images are analyzed to extract and normalize samples of facial parts such as mouth and eyes. Sections IV-A1 and IV-A2 detail the steps for finding the locations of facial parts and Section IV-B outlines the algorithms for the pose estimation. Using the pose information, the samples can be normalized and are then parameterized and stored in a database (Section IV-C). Once this database has been produced for a speaker, we can synthesize arbitrary new text, articulated by the talking head.

To synthesize speech animation, we start with a phonetic transcript of the text and search the database for mouth shapes corresponding to the given sequence of phonemes. Candidate mouth shapes are entered into an animation graph where an optimization algorithm picks the best mouth shapes for producing a smooth animation that is synchronized with the text. This step is discussed in Section IV-D. Having obtained mouth and eye animations in this way, we then use a 3-D model of these facial parts to find their correct appearance that can be overlaid onto a background face and rendered, either directly to the screen or to a video file (Section IV-E).

A. Image Analysis

Sample-based synthesis methods use recorded samples from which new video sequences can be generated. A crucial step in the production of a database of samples is the automatic analysis of the images in order to locate the positions of facial features, followed by their normalization. The latter step is necessary to remove unwanted variations in appearances. We try to remove the effect of rotation and translation of the head, so that all the extracted samples of facial parts appear to be viewed from exactly the same position and angle and to have the same scale. Analyzing images of faces is a very active field of research and a variety of algorithms have been developed over the years (see e.g., [85], [86]). The techniques described in the next two paragraphs proved to be computationally efficient and highly accurate, as well as robust against variations due to different complexions.

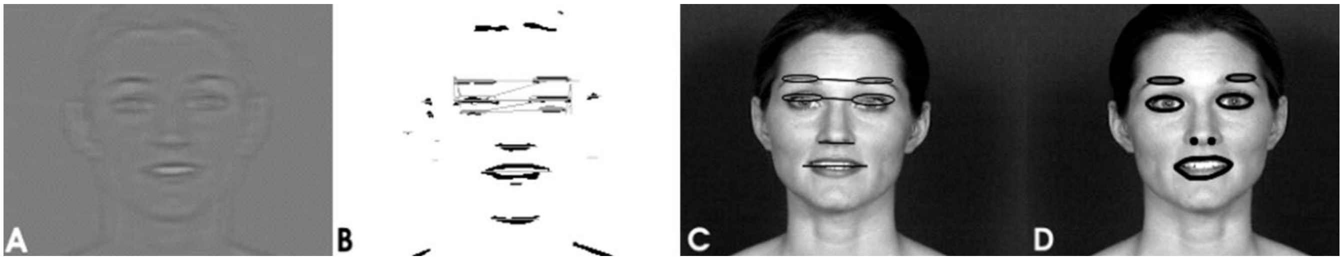


Fig. 8. Example of the processing steps to locate facial features. (A) Band-pass filtered image. (B) After morphological operations and thresholding; the light-gray lines show all the hypotheses tested for identifying the facial features. (C) The lines mark locations with the highest probability of representing facial features. (D) Approximate outline of facial features after color segmentation.

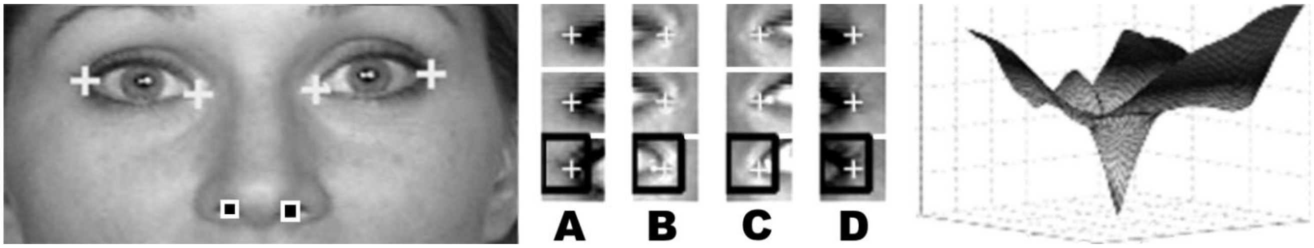


Fig. 9. Locating the corners of the eyes. For each corner there are three kernels, representing three levels of eye opening, for a total of 12 kernels, shown in the center (A, B, C, D). For the image on the left, the kernels corresponding to an open eye are selected (marked with dark boxes) and scanned over areas around the eye corners. The minima of the correlation functions mark the location of the corners (crosses in left image). The plot on the right-hand side shows the correlation function in a 60×60 pixel area around the left-most corner in the image on the left.

1) Locating Facial Features: We first need to locate within a video frame the position of the head, and then the location of the mouth and eyes. Typically, recordings for the database are produced in a studio environment with control over the background, the lighting, and the recording equipment. Hence, we are not dealing with difficult backgrounds or wildly varying scales and poses of the head. Under these conditions, finding the head and the approximate location of the facial features is not difficult. What makes this a challenging problem is that some facial feature points have to be identified with very high accuracy. We have to cut out a part of the face and overlay it onto another face. If the overlaid parts are offset by as little as one pixel, there will be visible artifacts in the animation. In order to find facial features with subpixel accuracy, we proceed in two steps, where first we locate several features only approximately and then zoom in to determine their locations more precisely.

The first step applies shape analysis and color segmentation as shown in Fig. 8. By filtering the image with a band-pass filter and applying morphological operations, we locate areas where facial features may be present. This produces a binary image [see Fig. 8(b)] where all the areas of interest are marked with black pixels. By analyzing the shapes and relative positions of these black areas, the ones marking eyes, eyebrows, and mouth are identified. Then, applying color segmentation to the mouth and eyes areas provides information about their shape (contour) [see Fig. 8(d)].

This analysis is reliable in identifying the facial features, yet is not very accurate in finding their precise locations. Typically, features are missed in less than 2% of the images and these can easily be inserted by interpolation between frames. Errors are caused mostly by movements that are so fast that the image becomes blurred. The locations of, for example, eye corners, as measured with this technique, may be several pixels off from their real positions. Filtering over time can improve these errors significantly, yet measurements that are more precise are still needed for the pose calculations.

2) High-Accuracy Feature Points: For calculating the 3-D head pose, we need to know at least four points in the face with high accuracy, preferably with an error of less than one pixel. We, therefore, add an additional level of analysis to measure a few feature points with the highest accuracy. From the data set analyzed as described above, a few representative examples of one feature point are selected. For example, for measuring the position of the left corner of the left eye, three image samples are selected (see Fig. 9). These samples are chosen based on how widely open the left eye is. This process can be automated, since width and height of the eyes have been measured with color analysis. From those images, the areas around the left corner are cut out. For analyzing a new image, one of these sample images is chosen, namely, the one where the width and height are most similar, and it is then scanned over an area around the left half of the left eye. We typically use pixel-by-pixel multiplication or difference on a band-pass filtered version of the images. This correlation identifies very precisely, where a feature point is located. The standard deviation of the measurements is typically less than one pixel for the eye corners, and filtering over time reduces the error to less than 0.5 pixels.

This technique has been chosen for its robustness and simplicity, as well as for its high accuracy. Only a small number

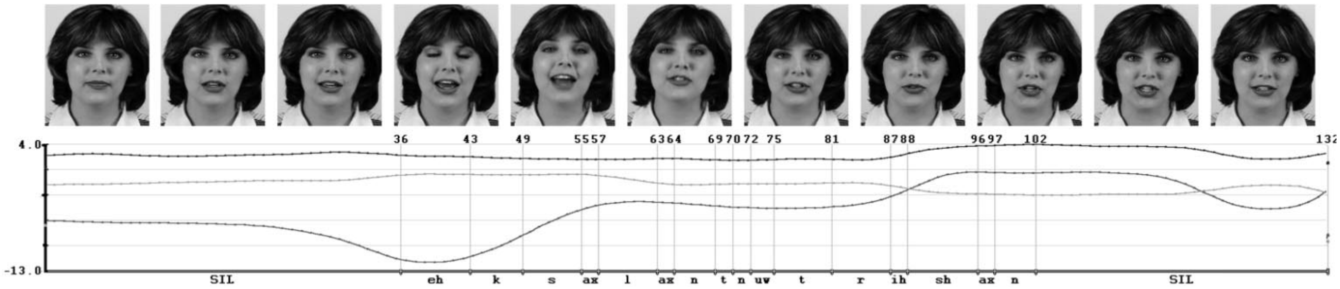


Fig. 10. Pose estimation example: bottom curve: rotation angle around the x axis (yaw); middle curve: rotation around the y axis (pitch); top curve: rotation around the z axis (roll). Vertical axis: degrees; horizontal axis, top: frame numbers, bottom: phonemes. Total duration of the utterance: 2.2 seconds.

of training images are needed to determine good values of algorithmic parameters, such as the kernel sizes. The correlation is computationally expensive, but the correlation function around the minimum tends to be smooth and can be approximated well by a parabolic function (see Fig. 9). Therefore, if computation time is an issue, optimization algorithms can be applied, such as Newton's algorithm. Furthermore, since the color segmentation provides already a good estimate of the eye corners' positions, we only need to scan over a small area. Scanning the areas around all four eye corners takes less than 300 ms on a 900-MHz PC, so that, typically, we do not bother with any further speed optimizations. More details about the image analysis can be found in [87].

B. Pose Estimation

During the recording, we also acquire the shape of the subject's head by measuring the relative positions of a few features (the four eye corners and the nostrils) on two calibrated images taken from different directions (frontal and side view). In this way, we build a simple 3-D model of the head. Measuring a few more points, we then construct the *approximate* model of the mouth area and the area around the eyes (these 3-D models consist of about 15 polygons each). More recently, we have used a Cyberscanner to acquire the geometry of a subject's head. This approach provides better accuracy and makes the procedure of capturing the 3-D position of feature points and the 3-D shape of facial parts more automated.

Knowing the values of both the 3-D feature points and, from the image analysis step, their corresponding 2-D points in the image plane and using the equations of perspective projection, one can infer the pose of the object. This is known as the perspective-n-point (PnP) problem. Several algorithms have been reported that solve it using the Newton–Raphson method [88], [89]. All are computationally expensive, may be slow to converge, and require an initial guess. We use instead a pose estimation technique reported in [90] and [91]. In this method, the pose is first estimated using a scaled orthographic projection, and then it is iteratively refined to match a perspective projection. Given that the 3-D feature points can be measured with high precision, the precision of the pose estimation is limited by the precision with which the 2-D feature points have been measured. At a resolution of about 50 dpi, a one-pixel displacement in one 2-D feature results in a change of less than 2° in the estimated pose (average over all possible displacements of six features). Because of controlled recording conditions (lighting, background, cooperating subjects), we obtain subpixel accuracy in the 2-D feature locations, and because of temporal coherence, we can further increase accuracy in the pose estimation. An example showing the results of pose estimation is shown in Fig. 10.

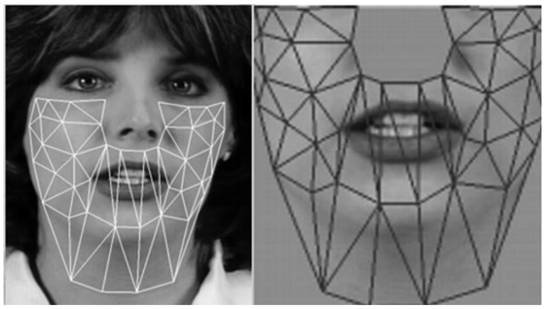


Fig. 11. Sample extraction and normalization. Left, the 3-D model of the mouth is projected onto an image using the head pose information. Right, the same 3-D model is projected using a normalized frontal pose. Standard texture mapping techniques are used to transform the pixels.

Other techniques that use optical flow constraints coupled with generic 3-D head models have been reported [92], [93] that also solve the pose estimation problem. However, in our case, since we know precisely the 3-D position of a few salient features (measured on the subject by a Cyberscanner or on calibrated images) as well as their 2-D positions (from the image analysis), we can afford this simpler, faster approach.

C. Normalization and Parameterization of Image Samples

Before the image samples are entered into the database they are corrected in shape and scale to compensate for the different head orientations they had when they were recorded. Using the pose, 3-D facial parts are projected onto the sample images and the texture is extracted and reprojected using a frontal pose. Fig. 11 illustrates this process.

Any information about the shape produced by the recognition module is also mapped into the normalized view and stored alongside the bitmap in a data structure. Once samples of a face part are extracted from the video sequences and normalized, they need to be labeled and sorted in a way that they can be retrieved efficiently. The first parameter used to describe a mouth shape is the phoneme sequence spoken during the recording. The speech audio data of all recorded sequences is segmented into phonemes by a speech recognition system using forced alignment (commercially available systems exists to perform alignment in a speaker independent [70] or a trained, speaker-dependent [94] mode) and the segmentation is double-checked by a human listener. A second set of parameters are the measurements produced by the facial feature recognition system such as mouth width and height.

While the geometric features just described are useful to discriminate among a large number of samples of a given facial part, they cannot capture fine details of the appearance. Wavelet representation, PCA, and linear discriminant analysis (LDA) have been used successfully for lip reading [95] and have been demonstrated to capture reliably the appearance of mouth and lip images. Here we choose PCA, since it provides a compact representation and captures the appearance of the mouth with just a few parameters. Typically about 15 coefficients account for over 95% of the variance in the mouth images, which is sufficient for defining a distance between samples. Misalignment of samples (offsets, rotations, etc.) would prevent PCA from accounting for most of the variance with a small number of principal components. However, since the samples have been normalized (thus factoring out the head pose), PCA components are good features for discriminating fine details in the appearance of facial parts.

Additional features stored in the database include the phonetic context, the head pose, and the sequence and frame number in the original recorded video data. This information is used when selecting samples for an animation in order to preserve, whenever possible, the inherent smoothness of real, recorded facial movements.

D. Unit Selection

Unit selection is the process of selecting units (or samples) from the database of recorded samples and assembling them in an optimal way to produce the desired target. The approach shown here is similar to techniques developed in TTS synthesis for producing natural-sounding synthesized voices [2], [3], [96].

Our system starts from a phonetically labeled target text. As described in Sections II-B and II-C, this target can be produced either by a TTS system or by a labeler or an aligner from recorded audio. From the phonetic target, we build an animation graph with n states corresponding to the n frames of the final animation. Each state of the final animation (one video frame) is populated with a list of candidate nodes (a recorded video sample from the database). Each state is fully connected to the next and *concatenation costs* are assigned for each arc, while *target costs* are assigned to each node. Fig. 12 illustrates the structure of such a graph. A Viterbi search through the graph finds the optimum path (the one that generates the lowest total cost).

The task of the unit selection is to balance two competing goals. The first goal is to ensure lip synchronization. Working toward this goal, the *target cost* uses phonetic and visemic context to select candidates that most closely match the phonetic and visemic context of the target. The target feature vector at frame t

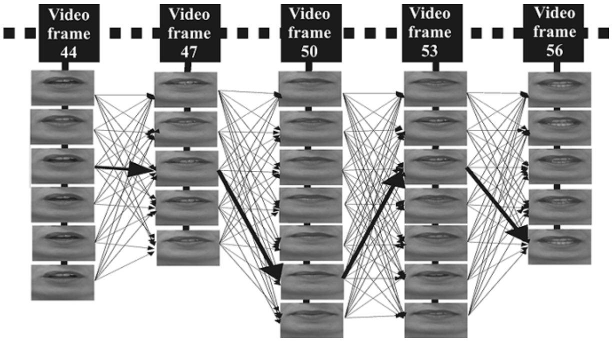


Fig. 12. A slice of a synthesis graph. For each frame of the final animation, a node is created in the synthesis graph. Each node is populated with a variable number of candidates during the unit selection process. Candidates of a node are fully connected to the candidates of the following node. The graph represents all possible animations that can be created by concatenating one candidate at each node. By assigning costs to arcs and using a Viterbi search algorithm, an optimal path [as defined by the cost function(s)] can be determined to produce a desired animation.

$$\boldsymbol{T(t)} = \{ph_{t-nl}, ph_{t-nl-1}, \ldots, ph_{t-1}, \\ ph_t, ph_{t+1}, \ldots ph_{t+nr-1}, ph_{t+nr}\}$$

is of size $nl + nr + 1$, where nl and nr are, respectively, the extent in frames of the coarticulation left and right of the target ph_t (the phoneme being spoken at frame t). A weight vector of the same size is given by

$$\boldsymbol{W(t)} = \\ \{w_{t-nl}, w_{t-nl-1}, \ldots, w_{t-1}, w_t, w_{t+1}, \ldots w_{t+nr-1}, w_{t+nr}\}$$

where $w_i = e^{-\alpha|t-i|}$, $i \in [t - nl; t + nr]$.

This weight vector simulates coarticulation by giving an exponentially decaying influence to phonemes, as they are further away from the target phoneme. Typical value for nr, nl, α are 10, 10, 0.3.

For a given target and weight vector, the whole database is now searched to find the best candidates. A candidate extracted from the database at frame u has a feature vector

$$\boldsymbol{U(u)} = \{ph_{u-nl}, ph_{u-nl-1}, \ldots, ph_{u-1}, \\ ph_u, ph_{u+1}, \ldots ph_{u+nr-1}, ph_{u+nr}\}.$$

It is compared with the target feature vector. The target cost for frame t and candidate u is given by

$$TC\,(t,u) = (1/\sum_{i=-nl}^{nr} w_{t+i}) \sum_{i=-nl}^{nr} w_{t+i} \cdot M(T_{t+i}, U_{u+i})$$

where $M(ph_i, ph_j)$ is a $p \times p$ "phoneme-to-phoneme visual distance matrix" where p is the number of phonemes in the alphabet. This matrix denotes visual similarities between phonemes. For example, the phonemes $\{m, b, p\}$, while different in the acoustic domain, have a similar visual appearance. This matrix is calculated using the database of recorded mouth images. Each phoneme is assigned an average visual feature vector obtained by averaging the feature vectors of all mouth images that were labeled with that phoneme. The composition of the feature vector is described later in the definition of the transition cost. The visual distance between two phonemes is then simply the Euclidian distance in feature space.

The second goal is to ensure smoothness in the final animation. An animation is a concatenation of segments from the database. When a segment ends with a frame that is visually different from the start frame of the following segment, a jerk occurs resulting in a less pleasing animation. The solution resides in both reducing the number of segments in a given animation and reducing the visual difference between segments. The *concatenation cost* works toward these goals using two distinct components: the *skip cost* and the *transition cost*. The skip cost $g(u1, u2)$ is null for two consecutive candidate frames that were recorded in that order and non-null otherwise. See equation at the bottom of the page.

The transition cost reflects the visual difference between two frames. Each frame i is assigned a feature vector U_i and the visual distance $f(U_i, U_j)$ is given by the Euclidian distance in the feature space. The feature vector is composed of eight weighted normalized features. For the first two features, we use the height and the width of the mouth as measured by the face analysis module described in Section IV-A. The next six features are the first six principal components of a PCA done on the entire set of normalized mouth images.

The concatenation cost is then given by $\mathrm{CC}(u1, u2) = f(U1, U2) + g(u1, u2)$. The graph $G = \{S_0, S_1, \ldots, S_n\}$ with states S_i and candidates $S_{i,j}$ can be constructed with a target cost TC for each candidate and a concatenative cost CC for each arc going from state S_i to state S_{i+1}. A path $\{p_0, p_1, \ldots, p_n\}$ through this graph generates the following total cost:

$$c = \mathrm{WTC} \cdot \sum_{t=0}^{n} \mathrm{TC}(t, S_{t,p_t}) + \mathrm{WCC} \cdot \sum_{t=1}^{n} \mathrm{CC}(S_{t,p_t}, S_{t-1,p_{t-1}}).$$

The best path through the graph is the path that produces the minimum cost. The weights WTC and WCC are used to fine-tune the emphasis given to concatenation cost versus target cost, or, in other words, to emphasize acoustic versus visual matching. A strong weight given to concatenation cost will generate very smooth animations but the synchronization with the speech might be lost. A strong weight given to the target cost will generate an animation which is perfectly synchronized to the speech but that might appear visually choppy or jerky due to the high number of switches between database sequences.

A good indicator of the quality of the synthesized animation is the average length of the segments chosen from the database. An average length of less than 100 ms indicates transitions every three frames or less and results in a visually choppy animation. We found empirically that an average segment length of 200 to 300 ms produces the best results with visually smooth and well-synchronized animations. Weights WTC and WCC were determined through a training procedure where 20 sentences were generated repeatedly with different weights [97].

E. Rendering

Once the best path through the animation graph has been calculated, an animation "script" is obtained that designates, for each frame of the animation, which particular image sample has to be retrieved from the database. Because of the effect of the skip and concatenation costs, the animation script is made of segments of consecutively recorded image samples. The Viterbi search has done its best to minimize visual differences at the junction of these segments; however, because the database has a limited size, many junctions remain for which a large visual difference exists. These abrupt transitions are generally noticeable on the final animation and greatly lower its overall quality. We use gradual blending of the image samples to smooth the animation over these junctions. The blending factor gradually blends a segment into its neighbor over a series of k frames. The length k of the transition is proportional to the visual distance at the junction of the segments but is bounded in duration by $\min$ and $\max$ parameters.

Our system renders directly either to the screen via OpenGL or to a video file. For real-time applications, screen rendering is necessary. Thanks to extensive optimizations, our system is able to keep up with the demands of real-time rendering on the screen at 30 frames/s. We use a 3-D textured mesh for the "base head" and video textures for animated facial parts. Fig. 13(a) illustrates this rendering process. To reduce latency for real-time applications, the unit selection is accelerated by selective graph pruning. This is achieved by allowing more candidates when their target cost is high (phonetic mismatch). Conversely, when their target cost is low, the chance to have a good candidate is higher and the need for more alternative candidates is lessened; thus, the

$$g(u1, u2) = \begin{cases} 0 & \text{when } |fr(u1) - fr(u2)| = 1 \text{ AND } \mathrm{seq}(u1) = \mathrm{seq}(u2) \\ w_1 & \text{when } |fr(u1) - fr(u2)| = 0 \text{ AND } \mathrm{seq}(u1) = \mathrm{seq}(u2) \\ w_2 & \text{when } |fr(u1) - fr(u2)| = 2 \text{ AND } \mathrm{seq}(u1) = \mathrm{seq}(u2) \\ \cdots & \\ w_{p-1} & \text{when } |fr(u1) - fr(u2)| = p - 1 \text{ AND } \mathrm{seq}(u1) = \mathrm{seq}(u2) \\ w_p & \text{when } |fr(u1) - fr(u2)| \geq p \text{ OR } \mathrm{seq}(u1) \neq \mathrm{seq}(u2) \end{cases}$$

where $0 < w_1 < w_2 < \cdots < w_p$, $\mathrm{seq}(u) = \mathrm{recorded\_sequence\_nb}$ and $\mathrm{fr}(u) = \mathrm{recorded\_frame\_nb}$.

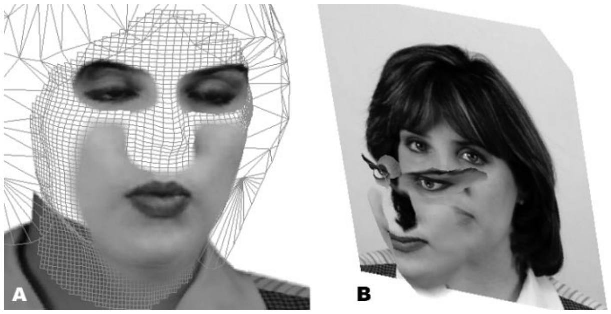


Fig. 13. (A) Rendering the head with a 3-D mesh model of the "substrate head." Eyes and mouth texture maps are changed for each frame, while the rest of the head is rendered always with the same texture. The "substrate head" model is shifted and rotated to add head movements. (B) Rendering of the head with a 2-D texture map as "substrate head." Eyes and mouth texture maps are overlaid onto a video sequence of the head. The pose of the "substrate head" is measured to place the facial parts properly.

graph can be pruned. More details on this technique can be found in [97].

Alternatively, we can improve the overall quality of the rendered animation by using a 2.5-D method where the "base-head" is also a recorded video sequence. Facial parts are then overlaid using head pose information, as illustrated on in Fig. 13(b). This approach tends to produce more natural-looking animations than the rendered 3-D head. However, the head cannot be shown in arbitrary pose. Rather, we have to search the database for sequences of head movements that fit the animation.

V. Visual Prosody

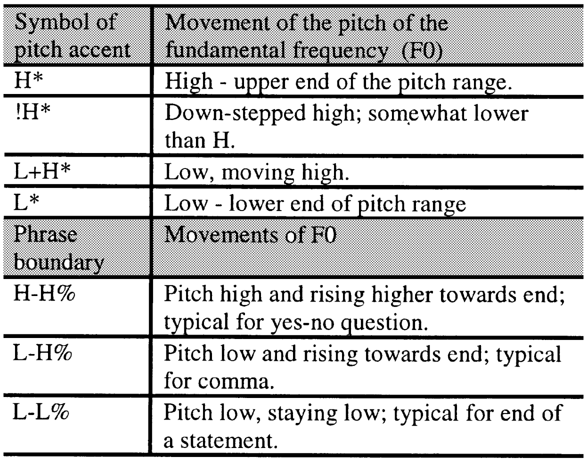
A. Speaking is More Than Moving the Lips

Prosodic movements are key for an animation to appear natural. For example, random movements or repetitive motion patterns appear unnatural and are easily identified as synthetic. In animations where the head moves randomly, it seems to float over the background, which is judged by most viewers as "eerie." If we drive the head with *recorded* movements, it looks more natural, even if the movements are *not* related with the text. Yet, truly natural appearance is obtained only if head and facial movements are synchronized with the text. How to obtain such synchronized head movements is described in this section.

Notice that we focus on movements that can be predicted from the syntax and the prosody of the text alone and do not rely on any semantic information. Predicting the prosody from text is one of the major tasks for TTS synthesizers; therefore, reliable tools exist. Analyzing the text for semantic meaning, on the other hand, is unreliable and may lead to embarrassing errors, such as a smile while articulating a sad message. Therefore, we rely on annotations done by humans for any information regarding emotions. If no such information is available, only very subtle smiles are added every now and then at the end of sentences, in order to enliven the animations.

The approach described here is, again, sample-based, where we observe the behavior of several speakers and collect a set of characteristic motion patterns. These motion segments are then concatenated to synthesize new animations. More details are available in [56]. It has to be noted that all the data described here were recorded while the speaker was reading from a teleprompter and the movements apply to this situation. How well these head movements describe other situations, for example, face-to-face conversations, remains to be investigated.

Table 1
ToBI Symbols for Marking Pitch Accents and Phrase Boundaries



Symbol of pitch accent	Movement of the pitch of the fundamental frequency (F0)
H*	High - upper end of the pitch range.
!H*	Down-stepped high; somewhat lower than H.
L+H*	Low, moving high.
L*	Low - lower end of pitch range
Phrase boundary	Movements of F0
H-H%	Pitch high and rising higher towards end; typical for yes-no question.
L-H%	Pitch low and rising towards end; typical for comma.
L-L%	Pitch low, staying low; typical for end of a statement.

B. Prosody

The prosodic phrase boundaries and pitch accents were extracted from a database of 5 h of recorded video with six different speakers. The head movements are measured in all these videos with the image analysis tools described above. The prosodic events are labeled according to the Tones and Break Indices (ToBI) prosody classification scheme [98]. ToBI labels do not only denote accents and boundaries, but also associate them with a symbolic description of the pitch movement in their vicinity. The symbols shown in Table 1 indicate whether the fundamental frequency ($F0$) is rising or falling. The two-tone levels, high (H) and low (L), describe the pitch relative to the local pitch range and baseline.

C. Prosodic Head Movements

For identifying prosodic head movements, the three angles of rotation together with three translations are measured. All the recorded head and facial movements were added spontaneously by the speakers while they were reading from a teleprompter. The speakers were not aware that their head movements would be analyzed. For most of the recordings, the speakers were asked to show a "neutral" emotional state.

For the analysis, each of the six signals representing rotations and translations of the head is split into two frequency bands:

0–2 Hz: Slow head movements
2–15 Hz: Faster head movements associated with speech.

Movements in the low frequency range extend over several syllables and often over multiple words. Such movements

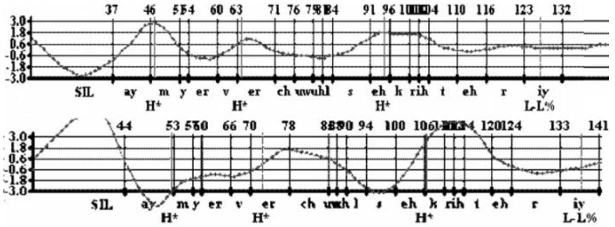


Fig. 14. Head rotation angle around the x axis (pitch) as a function of time, while articulating the sentence "I'm your virtual secretary." Shown on the horizontal axes are the phonemes and prosodic events. Clearly visible are the nods placed on the prosodic events. Top: Articulated with a neutral emotional expression. Bottom: Articulated with a joyful expression.

tend to be caused by a change of posture by the speaker, rather than being related to prosodic events in the speech.

The faster movements, on the other hand, are closely related to prosody. Accents are often underlined with nods that extend typically over two to four phones. This pattern is clearly visible in Fig. 14 where the nods are very clearly synchronized with the pitch accents (positive values for angle $\boldsymbol{ax}$ correspond to down movements of the head). Typical for visual prosody, and something observed for most speakers, is that the same motion—in this case, a nod—is repeated several times. Not only are such motion patterns repeated within a sentence, but often over an extended periode; sometimes over as much as the whole recording session, lasting about half an hour.

A further characteristic feature of visual prosody is the initial head movement, leading into a speech segment after a pause. In Fig. 14 this is shown as a slight down movement of the head ($\boldsymbol{ax}$ slightly positive), followed by an upward nod at the start of the sentence. We recorded 50 sentences of the same type of greetings and short expression in one recording session. The speaker whose record is shown in Fig. 14 executed the same initial motion pattern in over 70% of these sentences.

In Fig. 14, only the rotation around the x axis (yaw) is shown. In this recording, the rotation $\boldsymbol{ax}$, i.e., nods, was by far the strongest signal. Many speakers emphasize nods, but rotations around the y axis are quite common as well, while significant rotations around the z axis are less common. A combination of $\boldsymbol{ax}$ and $\boldsymbol{ay}$, which leads to diagonal head movements, is also observed often.

The mechanics for rotations around each of the three axes are different; consequently, the details of the motion patterns vary somewhat. Yet, the main characteristics of all three of these rotations are similar and can be summarized with three basic patterns.

1) Nod, i.e., an abrupt swing of the head with a similarly abrupt motion back.
2) Nod with an overshoot at the return, i.e., the pattern looks like an "S" lying on its side.
3) Abrupt swing of the head without the back motion. Sometimes the rotation moves slowly, barely visible, back to the original pose, sometimes it is followed by an abrupt motion back after some delay.

Table 2
Percent of Pitch Accents Accompanied by a Major Head Movement. Text Corpus: 100 Short Sentences and Greetings, One Speaker

$P(\wedge_x \mid *)$	42 %
$P(\sim_x \mid *)$	18 %.
$P\,(/_x \mid *)$	20%

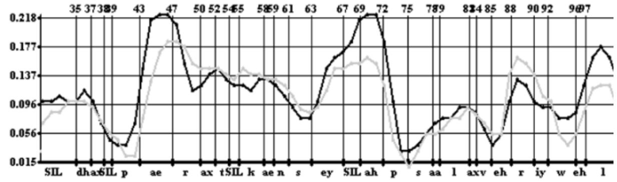


Fig. 15. Mouth height as a function of time during the articulation of the sentence, "The parrot can say Uppsala very well." Light gray curve: synthesized articulation; black curve: recorded video. Notice the transitions "p-ae" and "ah-p"; they represent extreme transitions from open to closed and the reverse. Key for a good synchronization is to get such transitions at the right time. The precise shapes of the openings are less important for the perceived quality of the articulation.

We can summarize these patterns with three symbols, where each one can be executed around the x, y, or z axis:

$\wedge$ nod (around one axis)
$\sim$ nod with overshoot
$/$ abrupt swing in one direction

Having such a basis set of motion primitives allows describing head movements with the primitives' types, amplitudes, and durations. This provides a simple framework for characterizing a wide variety of head movements with just a few numbers. Table 2 shows some statistical data of the appearance of these primitives in one part of the video database.

The amplitudes of the movements can vary substantially, as illustrated by Fig. 14, bottom. For this recording, the speaker was asked to articulate the same sentence as in Fig. 15, top, but with a cheerful expression. The initial head motion is now a wide down and up swing of the head, which runs over the first nod seen in the upper figure. The first nod falls now on the second accent and the sentence ends with an up–down swing.

The patterns described here are not always visible as clearly as in the graphs of Fig. 14. Some speakers show far fewer prosodic head movements than others. The type of text being read also influences prosodic head movements. When reading paragraphs from the *Wall Street Journal*, the head movements were typically less pronounced than for the greeting sentences. On the other hand, when speakers have to concentrate strongly, while reading a demanding text, they often exhibit very repetitive prosodic patterns.

Head and facial movements during speech exhibit a wide variety of patterns that depend on personality, mood, content of the text being spoken, and other factors. Despite large variations from person to person, patterns of head and facial movements are strongly correlated with the prosodic structure of the text. Angles and amplitudes of the head movements vary widely, yet their timing shows surprising consistency. Similarly, rises of eyebrows are often placed at prosodic events, sometimes with head nods, at other times without.

Visual prosody is not as predictable as acoustic prosody, but is clearly identifiable in the speech of most people. Adding prosodic head movements to an animation can make a big difference in the appearance of a talking head. While without it, the head appears disengaged and robot-like, when it is added, visual prosody creates the illusion that the head understands what it articulates. Having a basis set of motion primitives plus the statistics of frequency and amplitude of these movements provides a framework to drive a talking head. Different personalities can be emulated by using the statistics of a particular person having that personality.

With a 3-D head model, it is straightforward to apply the head movements that emulate visual prosody. For a sample-based approach, we use recorded segments of prototypical head movements that are concatenated. This can be done with a relatively small number of sample movements, since the timing precision of these movements does not have to be very precise. In order to find matching sequences for a new sentence, the prosodic events produced by TTS are compared to those of recorded sentences in the database. We assume that if the acoustic prosody is similar, the visual prosody from the recorded sentence is going to be a good match for the new sentence. The prosodic similarity between two sentences is measured by a string matching operation, comparing the locations of stress marks. If two stress marks in the two different sentences line up perfectly their distance is zero. Differences in time increase the distance, as do insertions and deletions. Using this metric, the best matching sentence from the database is selected as background and the synthesized lip sequence is overlaid.

VI. Quality Assessment

Assessing the quality of an animation is key for making any progress and becomes even more urgent as the animations become more lifelike, since improvements may be more subtle and subjective. Subjective testing, where human observers provide feedback is the ultimate measure of quality. However, this is time consuming and expensive, since we need a large number of observers, preferably from different demographic groups. Usually, many of the complaints of viewers can be explained by measurable characteristics of the animations. An automatic quality assessment aims at identifying such characteristics and tries to quantify their influence on quality. Automatic quality assessment will never replace subjective tests, but can greatly accelerate the development and also increases the efficiency of subjective tests by focusing them on the important issues. We discuss both methods.

A. Automatic Quality Evaluation

A good way of characterizing the quality of an animation is to synthesize a sentence that has also been recorded, preferably articulated by the same person that has been recorded for the database. Fig. 15 shows such an example, where the mouth height as a function of time is plotted for the synthesized and the recorded sentence. By comparing 50 synthesized and recorded sentences and letting viewers judge such qualities as "naturalness," "smoothness," and "precision," we isolated a few measurable criteria that correlate strongly with the subjective impression of quality.

Closures: Experiments indicate that human viewers are very sensitive to closures, and getting the closures at the right time may be the single most important criterion for providing the impression that lips and sound are synchronized. Closures are easy to identify visually, simply by finding where the height of the mouth hits a minimum. For the plosives "b" and "p," the lips should be closed for the first part of the phonemes, since by the time the puff of air escapes the lips, making the audible sound, the lips are parting already. Unfortunately, labeling tools are often not very precise in finding the location and duration of "p" and "b," which can hamper in the quality measures. For the labiodental fricatives, "f" and "v," the visual impression of closure seems somewhat less significant, but for the bilabial "m," it is very important to have a visible closure of the lips.

Protrusions: Protrusions are visually prominent deformations of the lips and viewers are quite sensitive to an appearance of a protrusion with the vowels "ow," "uh," "uw," and the consonant "w." Identifying the onset and end of a protrusion is not easy to do automatically, but the ratio of lip width to height provides a good indication. The precise shape does not seem to matter as much as a visible reduction of the mouth width. A lack of protrusions makes the impression of slurred or careless articulation.

Turning Points: When the speaker's mouth changes the direction from opening to closing, or vice versa, this is marked as a turning point. These are typically points of high acceleration of the jaw and, therefore, of strong muscle action. The precise placement of a single turning point does not seem to be important, but if several in a row are different from those in a recorded sequence, viewers get the impression of poor synchronization.

Smoothness: If the mouth opens or closes too rapidly, this is perceived as jerky and unnatural. However, it has to be emphasized that people can close or open their mouth very rapidly (compare Fig. 15), and simply suppressing fast transitions looks very unnatural. Key for a natural appearance is that fast transitions are placed precisely. For example, the opening from a plosive consonant to a wide open vowel has to be synchronized well with the sound in order to look natural. In the sample-based synthesis, jerkiness is introduced mainly at the transitions where samples from different sequences are concatenated. Therefore, our automatic quality assessment focuses on visual differences across sequence boundaries as a metric of jerkiness. Jerkiness may also result from errors in the image analysis or the pose estimation.

Table 3 shows a comparison of objective, automatic quality scores and subjective scores, obtained from eight human observers. In order to judge the precision, the viewers were shown recorded and synthesized versions of the same sentences and asked to judge how precisely they match. They were also asked how well the sound is synchronized

Table 3
Comparison of Automatic (Objective) Scores and Subjective Scores; 20 Sentences Were Synthesized; for the Subjective Score Eight Viewers Judged the Quality of the Animations

	Automatic scores	Subjective scores
Precision	4.61	4.62
Synchronicity	4.39	4.78
Smoothness	2.48	4.29

with the lip articulation (synchronicity) and how smooth the articulation is (smoothness). The observers judged these criteria on a scale of 1 (bad) to 5 (excellent). For the automatic evaluation, precision was calculated by how far turning points lie from those in the recorded sentence. The synchronicity is based on how far closures and protrusions are from the centers of their respective phonemes and smoothness is calculated by summing the visual distance across segment boundaries. The objective metrics were tuned with a training set of ten sentences. As can be seen in Table 3, precision and synchronicity tend to agree relatively well with those of human observers. Smoothness is not captured that well with our criteria. Large differences in visual appearance across sequence boundaries are not judged as jerky if they appear in places where the observer expects fast mouth movements. Hence, the viewers do not necessarily judge this as jerky, despite large visual differences that lead to a low automatic score.

With these measurable criteria, we can determine immediately quality numbers for every animation. During algorithm tuning, we typically use 50 sentences that are being synthesized repeatedly with different values for the synthesis parameters. By comparing the quality numbers with those of the recorded sentences, we can identify the parameters providing the best animations.

B. Subjective Quality Tests

For subjective quality tests, viewers are presented with samples of animations and are asked to enter their evaluations, typically on a 1 (bad) to 5 (excellent) scale, resulting in a "mean opinion score" (MOS) [99]. For any subjective tests, it is important to separate carefully the different factors influencing the perceived quality of speech. For example, if the face looks pleasing, the head moves naturally, and smiles appear every now and then, the impression may be positive, even if the articulation is not well synchronized with the sound. Therefore, if viewers are to judge the quality of the articulation, head movements, and emotional expressions should be eliminated, or they have to be identical in the sequences that are being compared.

1) Passing the Turing Test: The ultimate goal is to produce animations that pass the Turing test, namely, that a viewer cannot distinguish between animation and recording. How difficult this is to achieve depends strongly on what the talking heads try to show. We categorize the tasks into four levels:

1) speech articulation, no other movements;
2) articulation with prosodic movements;
3) articulation with prosodic movements and emotions;
4) articulation with prosodic movements, emotions, and behavioral patterns.

Where the Turing test can be passed with today's technology is for task 1. In informal tests and in formal tests with 12 subjects asking whether a presented video with one sentence is synthetic or real, we have observed that four out of five animated videos are perceived as real videos. This is also supported by separately measured MOS scores. A formal test is reported in [33] where morphed articulations were compared to recorded videos. In one test, the viewers were shown an animation and asked to judge whether it is real. In a second test animation and recorded video were shown side by side. In both cases, the animations were indistinguishable from recordings. Video-realistic animations have been achieved only with sample-based techniques. To our knowledge, so far, no model-based 3-D heads have shown articulations with a quality comparable to recordings.

In the tests mentioned previously, the animated sentences are relatively short. In this case, it does not matter if the head movements are not synchronized with the speech, as long as they are small. Even no head movements are tolerable for such tests. However, as sentences become longer, or when the head articulates whole paragraphs, it is essential that proper head movements are added. To our knowledge no talking heads have been shown to emulate articulation plus head movements so that they could be mistaken for recordings. With the sample-based visual prosody described in Section V, we can produce animations with natural-looking head movements if we render the mouth with recorded "substrate heads" (see Section IV-E). If a 3-D "substrate head" is used, the animations tend to look artificial even if the head is driven with movements extracted from recordings. This phenomenon is widely observed in computer animation. Even when applying sophisticated motion capture equipment, animations based on 3-D models tend to look artificial.

Tasks 3 and 4 require a semantic interpretation of the text in order to introduce emotional expressions and behavioral patterns that appear meaningful. This is beyond present-day natural language understanding, and, for the time being, this requires interpretation by humans who then annotate the text with tags. Natural appearance can, in principle, be achieved with sample-based techniques. In practice, however, this may require such large databases of recorded samples that it may not be feasible beyond emulating some basic behaviors.

2) Improved Understanding: Sample-based and model-based 3-D heads are able to support lip reading by untrained observers. A number understanding test under noisy conditions with about 190 subjects revealed that the presence of a sample-based or a 3-D model head reduces the error rate of digit understanding by 50% [100]. Similar results are reported in [101]. This effect will increase the acceptance of talking faces in information kiosks located in noisy environments like airports or railway stations. A 3-D model can help hearing impaired people to understand syllables [102]. However, lip reading from a computer or TV screen with a talking face model or a real human remains extremely challenging even for trained lip readers. Therefore, if the support

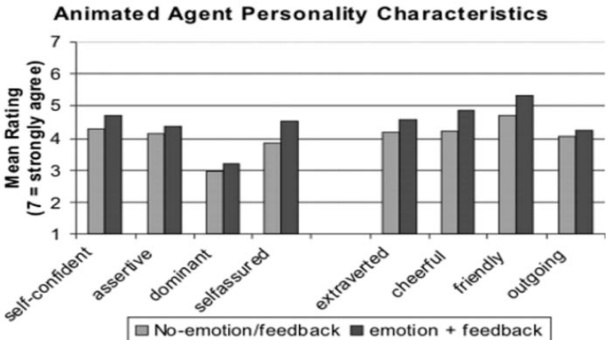
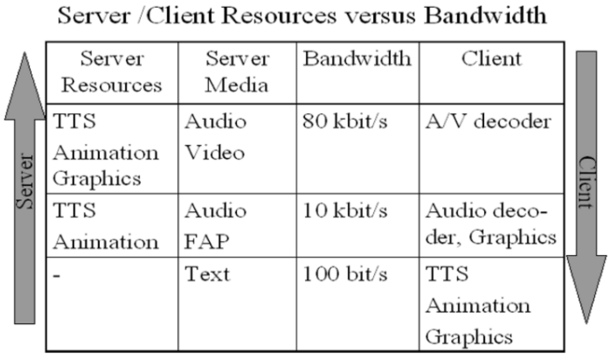


Fig. 16. Facial expressions appropriate for the dialogue situation improve the personality rating of a talking face in an interactive game. Mean rating 1 corresponds to "strongly disagree," 7 corresponds to "strongly agree." (Reproduced with permission from J. Ostermann, "E-COGENT: An electronic convincing agent," in *MPEG-4 Facial Animation: The Standard, Implementation and Applications*, I. S. Pandzic and R. Forchheimer, Eds. (Chichester, U.K.: Wiley, 2002) [18]).


Server
Client

Server /Client Resources versus Bandwidth

Server Resources	Server Media	Bandwidth	Client
TTS Animation Graphics	Audio Video	80 kbit/s	A/V decoder
TTS Animation	Audio FAP	10 kbit/s	Audio decoder, Graphics
-	Text	100 bit/s	TTS Animation Graphics

Fig. 17. Shifting resource requirements from the server to the client decreases the bandwidth requirements for animating the talking head. [Reproduced with permission from 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. (Chichester, U.K.: Wiley, 2002) [18]].

of hearing-impaired people is the main purpose of the talking face, the animation should also include the presentation of text on the screen.

Appropriate emotional feedback is important to engage the user more in an interaction [18]. Furthermore, a talking face with facial expressions is perceived as more likable. Users attribute more positive personality characteristics to a talking face if the face supports the semantics of its speech with facial expressions. Fig. 16 shows that the appropriate emotional feedback (joy and sadness in this example) improve the perceived personality of a 3-D talking face in an interactive game; that is, the talking face is perceived as more self-confident, assertive, cheerful, and friendly if it appears to be engaged in the dialogue with the user.

VII. Applications

We can imagine several architectures that enable Web-based face animation for interactive services via a local client. They differ in terms of software and hardware requirements for the client as well as bandwidth requirements for the server–client connection. Furthermore, the architecture determines the load each client puts on the server. Depending on the application, different architectures may be chosen. Fig. 17 shows the resources that a server requires for controlling a talking face in addition to a Web server and its control scripts. In a simple case, the server just sends the text to be spoken to the client. The client requires a TTS engine and a complete face animation system. If the TTS engine is available only at the server, the server needs to stream compressed audio and animation data like FAPs and optional phonemes and their duration to the client. The client requires an audio decoder and a graphics engine. If the server streams the phonemes, a coarticulation engine is also required at the client. If the server has a face animation system in addition to the TTS engine, it may render the entire animation on the server and stream audio and video to the client. This puts the least computational load on the client; however, the bandwidth requirements for reasonable video and audio quality are about 80 Kb/s. Therefore, this scenario is only suitable for broadband connections whereas the previous two scenarios can be used over mobile and dialup connections. Since the rendering of a talking face on a server is very compute intensive, this architecture is only viable for noninteractive applications like news broadcasting (Section VII-B) or e-mail services (Section VII-D). The dialogue system in Section VII-A, as well as the electronic store in Section VII-C requires use of one of the other two architectures. In Section VII-E, we discuss Internet protocols in order to enable Web-based interactive services with talking faces.

A. Help Desk

We used the scenario of a Help Desk application as a demonstration of our real-time player in a dialogue situation between a customer and a virtual customer service agent. It follows a client–server type of architecture and, hence, requires the installation of a talking head player. On the other hand, it allows very low bitrate communication between the server and the player because only animation parameters and audio (assuming that no TTS engine is available at the client) need to be sent. The client player is responsible for playing speech animations of the virtual customer service agent sent by the server and for capturing the user's input and sending it to the server. The server receives the user's input, interprets it, generates a response, computes the associated speech animation, and sends the animation parameters and audio to the client player. Fig. 18 shows the client–server architecture of the Help Desk application.

We use a natural language understanding (NLU) module to interpret the user's input [103], while the dialogue itself is directed by a dialogue management module [104]. These modules are part of a commercial system known as "How May I Help You?" [105], currently used by AT&T to run customer relation management (CRM) systems over phone lines.

To appear realistic in a dialogue situation, the virtual agent needs to exhibit idle and listening behavior while not speaking. Idle behavior includes mechanical facial motions, while listening behavior refers to a more complex

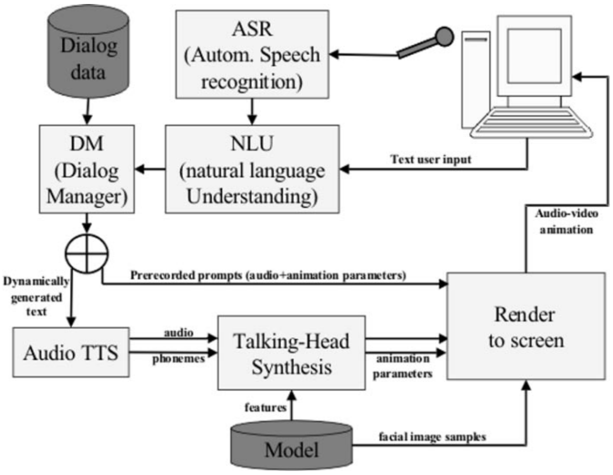


Fig. 18. Architecture of the "Help Desk" application. The user interfaces with the "Help Desk" application using either textual input (keyboard) or via a microphone (through an automatic speech recognition module). The latter has not yet been implemented (but all necessary components are readily available). The NLU module extracts structure and meaning from the raw text and ships the results to the dialogue manager, which in turn provides the textual answers for the agent based on dialogue data and state information. To reduce the workload on the server, the prompts that are known in advance are precomputed into animation parameter files. The dynamic content, on the other hand, has to be synthesized and, hence, is sent to the TTS engine and on to the talking head synthesis module which computes the animation parameters and sends them to the real-time renderer.

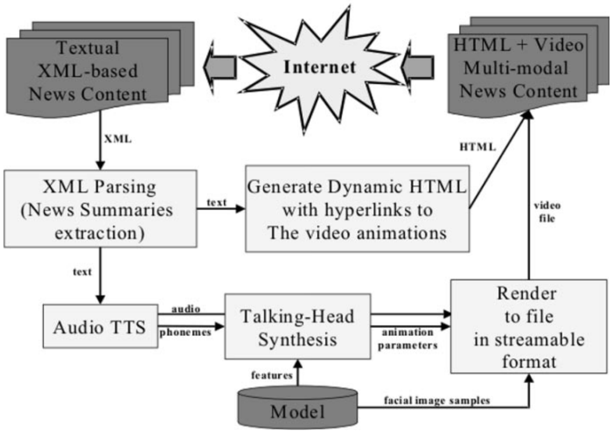
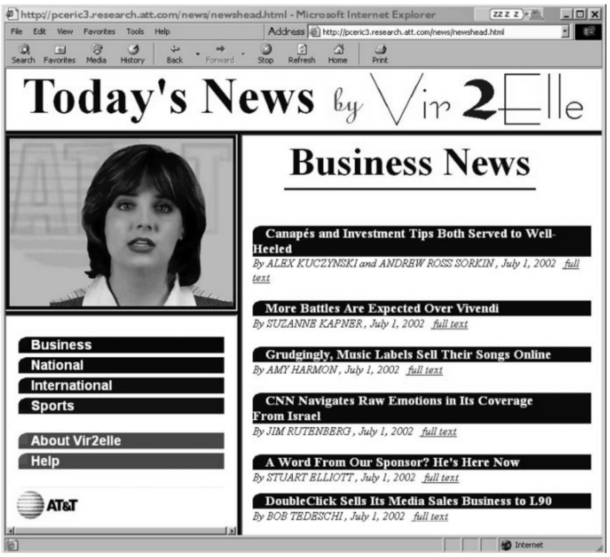

Textual XML-based News Content
Internet
HTML + Video Multi-modal News Content
XML
HTML
XML Parsing (News Summaries extraction)
text
Generate Dynamic HTML with hyperlinks to The video animations
video file
text
Audio TTS
audio
phonemes
Talking-Head Synthesis
animation parameters
Render to file in streamable format
features
Model
facial image samples

Fig. 19. Architecture of the automated newscaster application. News in XML format is obtained from the Web and parsed to extract structure and content. The raw text is then sent to the TTS module that produces the audio and the phonetic string used by the video synthesis module to compute the animation parameters. Finally, the renderer module synthesizes the animation into a streamable video file. The structure extracted from the XML document is used to build the final HTML page with links to the synthesized video files.


http://pceric3.research.att.com/news/newshead.html - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://pceric3.research.att.com/news/newshead.html
Search Favorites Media History Back Forward Stop Refresh Home Print
Business
National
International
Sports
About Vir2elle
Help
AT&T
Internet

Today's News by Vir2Elle

Business News

Canapés and Investment Tips Both Served to Well-Heeled
By ALEX KUCZYNSKI and ANDREW ROSS SORKIN , July 1, 2002 full text

More Battles Are Expected Over Vivendi
By SUZANNE KAPNER , July 1, 2002 full text

Grudgingly, Music Labels Sell Their Songs Online
By AMY HARMON , July 1, 2002 full text

CNN Navigates Raw Emotions in Its Coverage From Israel
By JIM RUTENBERG , July 1, 2002 full text

A Word From Our Sponsor? He's Here Now
By STUART ELLIOTT , July 1, 2002 full text

DoubleClick Sells Its Media Sales Business to L90
By BOB TEDESCHI , July 1, 2002 full text

Fig. 20. Screenshot of the Automatic Newscaster application. The Web site is organized using frames. On the left side of the screen, our talking head is speaking a summary of the headlines, while on the right side of the screen, the user can click on any headline and have the talking head directly skip to it or alternatively access the full text for reading. The talking head animation is generated entirely automatically from the textual content downloaded from the Internet.

set of facial motions and expressions used by protagonists during dialogues to indicate understanding or disagreement, facilitate turn taking and interruption, and suggest reflection or expectation. In its current implementation, however, the talking head player only supports idle behavior in the form of mechanical facial motions such as eye-blinks, eye motions, and small head movements.

B. News Reader

While our talking head player is capable of real-time synthesis using a dedicated player, it is sometimes desirable to animate talking heads on a client machine using other means. Dedicated players require download and installation and are often perceived as intrusive by users. An alternate way to distribute talking head animations over the Internet is to stream complete audio–video animations directly to client players such as Microsoft Windows Media Player, Apple Quicktime, or RealNetworks' Real Player. Most PCs are equipped with such players, making it a preferred way to distribute talking head content. We developed an automated newscaster application that produces multimedia content (video+HTML) that can be streamed to, and played on, client PCs.

The automated newscaster application periodically checks the Internet for news updates. After downloading headlines and summaries for sections such as business, technology, sports, etc., it dynamically builds a set of HTML Web pages that let the user browse the latest news in an engaging, interactive, and multimodal way. (A screenshot of the business news page from this site is shown in Fig. 20.) On the left side of the screen our talking head is speaking a summary of the headlines, while on the right side of the screen, the user can click on any headline and have the talking head directly skip to the related section, or alternatively access the full text for reading.

The talking head animation is generated entirely automatically from the textual content downloaded from the Internet. Fig. 19 shows the flow of operations involved in synthesizing a news animation from the downloaded textual content. HTML source is first obtained from the Internet and is then parsed to retrieve the relevant textual data. News providers on the Internet generally provide XML tags to identify logical parts within the news content, making the parsing robust against changes in the layout and style. The text needed to create the talking head animation is prepared for TTS synthesis by adding pauses between headlines and at the beginning and end of the animation. Later, emotions as well as music jingles will be inserted during these silences. The marked-up text is then sent to the TTS module and the resulting phoneme and audio blocks are used to calculate the visual animation.

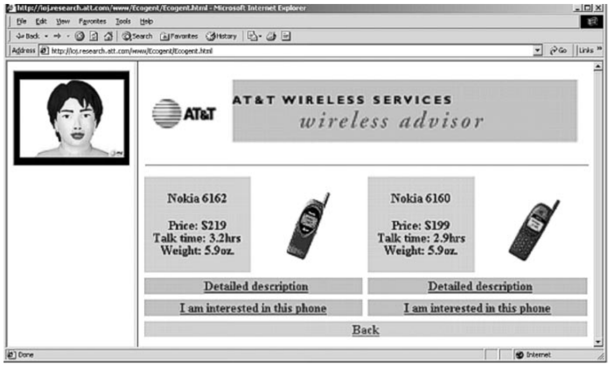
The animation is then rendered to memory and the image buffer is sent together with the audio to the video encoder, resulting in an encoded video file. The video file is placed on a media server and its reference is inserted in the dynamically generated Web pages. A view of such a page is shown in Fig. 20. To allow the user to choose any headline interactively, we produce as many separate video files as there are headlines. If the user does not select any headline in particular, the page plays all headlines. To allow for such flexibility, we use Synchronized Multimedia Interface Language (SMIL) [106] to play individual or chained-up video files.

C. E-Cogent

We implemented the "E-cogent" application, which helps customers choose a mobile phone. The customer is first asked a couple of questions regarding phone price, weight, and talk time. Then E-cogent presents available choices (see Fig. 21). The user may choose to see the detailed specifications of the phones, proceed to buying one, or go back to start over. In case the user starts over, he or she is presented with a screen as shown in Fig. 22. Compared to the first screen that the user saw, there is now the new button "Most popular phones." The talking head will verbally highlight this new choice to the user and recommend selecting that option. However, the user still has the choice to specify other preferences.

Initial attempts to integrate talking heads into existing dialogue systems failed to create a convincing experience because no effort was made to adapt the dialogue to the new modality of a talking head. A successful dialogue management system for talking heads needs to consider carefully what information is presented verbally and what information needs to be printed on the screen. In the following, we list a couple of guidelines for a dialogue that involves a talking head. These guidelines are not meant to be exhaustive but represent a snapshot of our current knowledge.

1) The talking head must appear alive, i.e., it has to move while not speaking. When waiting for input, the talking head should have a listening posture.
2) The talking head may under no circumstances slow down or interfere with the interaction with the user. To that extent, it appears useful to design the Web site such that it is functional even without the talking head.
3) The talking head has to speak whenever a new Web page is presented. However, it should not read what is written on the screen. It should emphasize highlights and avoid long sentences. For example, if the user requests technical details for a phone in Fig. 21, the talking head will merely say something like "Here are the technical details." For Fig. 22, we use a comment like "You are back here. Perhaps you would like to see our most popular phones."


http://loj.research.att.com/www/Ecogent/Ecogent.html - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://loj.research.att.com/www/Ecogent/Ecogent.html
Go Links
AT&T
AT&T WIRELESS SERVICES
wireless advisor
Done
Internet

Nokia 6162

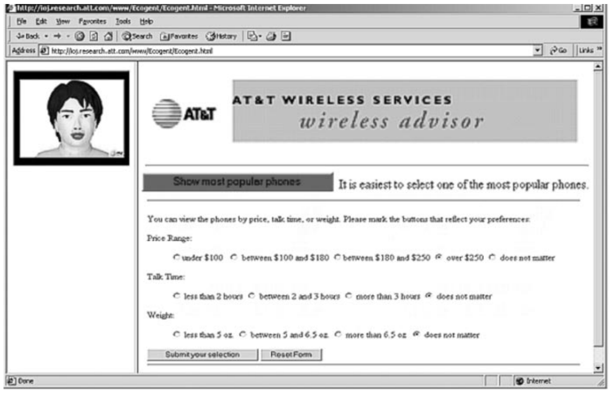
Price: $219
Talk time: 3.2hrs
Weight: 5.9oz.

Nokia 6160

Price: $199
Talk time: 2.9hrs
Weight: 5.9oz.

Detailed description
Detailed description
I am interested in this phone
I am interested in this phone
Back

Fig. 21. E-cogent presents several phones that fit the needs of the customers. [Reproduced with permission from 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. (Chichester, U.K.: Wiley, 2002) [18]].


http://loj.research.att.com/www/Ecogent/Ecogent.html - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://loj.research.att.com/www/Ecogent/Ecogent.html
Go Links
AT&T
AT&T WIRELESS SERVICES
wireless advisor
Show most popular phones
Submit your selection
Reset Form
Done
Internet

It is easiest to select one of the most popular phones.

You can view the phones by price, talk time, or weight. Please mark the buttons that reflect your preferences:

Price Range:

○ under $100 ○ between $100 and $180 ○ between $180 and $250 ● over $250 ○ does not matter

Talk Time:

○ less than 2 hours ○ between 2 and 3 hours ○ more than 3 hours ● does not matter

Weight:

○ less than 5 oz. ○ between 5 and 6.5 oz. ○ more than 6.5 oz. ● does not matter

Fig. 22. E-cogent verbally highlights the new button on the page that leads to the most popular phones. [Reproduced with permission from 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. (Chichester, U.K.: Wiley, 2002) [18]].

4) The talking head may not repeat the same sentence repeatedly. The server has to keep state information of the dialogue. This enables the server to choose different sentences with identical meanings when presenting the same Web page again.
5) Monitoring the click trail enables the server to detect easily customer frustration and issue sentences and facial expressions that soothe the customer.
6) The server may make the talking head act like a regular sales person. As such, the talking head can also engage in up-sale techniques.

These guidelines indicate that a talking head might be most efficiently used to guide a customer to getting the information and service he or she is looking for. Finally, talking heads may regain the attention of the customer even when the customer is not looking at the screen anymore.

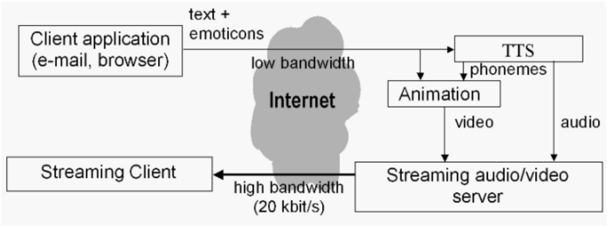


Fig. 23. Overview of the PlayMail architecture with the client and the server connected via the Internet. [Reproduced with permission from J. Ostermann, "E-COGENT: An electronic convincing agent," in *MPEG-4 Facial Animation: The Standard, Implementation and Applications*, I. S. Pandzic and R. Forchheimer, Eds. (Chichester, U.K.: Wiley, 2002) [61]].

D. Playmail

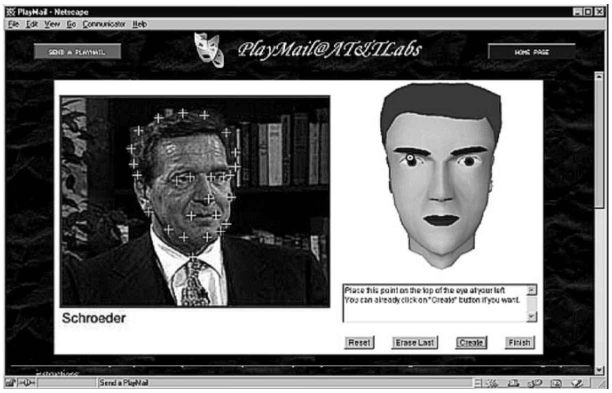
PlayMail is a multimedia enhanced e-mail service that translates text messages into animated videos with face models reading the message [61], [107]. The face models are created, customized, and selected by the user. Using a Web-based interface, the user creates a face model from a frontal picture of a friend, teacher, or celebrity that he or she submits into his/her private or public database of face models. In order to communicate emotions, the sender may use several predefined emoticons like :-) for smile or :-(for frown in the text. The PlayMail recipient will receive a short e-mail with a link to a Web page that has the message video embedded using Real Player [108] or the Windows Media Player [109].

Fig. 23 shows the system architecture of the PlayMail service. The client sends an http-request with the information for sending a PlayMail to the server. Since the server may not be able to compute all PlayMails immediately, it places all requests in a first-in, first-out queue. For each request, the server renders the video of the face animation with the user-selected face model using its face animation engine. The streaming audio and video server stores the PlayMail message for retrieval at different bit rates in order to accommodate recipients with different network access speeds. Finally, the server sends a regular e-mail to the recipient providing the URL of the PlayMail message.

When retrieving the PlayMail message, the server probes the connection to the client and starts streaming at the highest available bandwidth that can be sent to the receiver. The bandwidth gets adapted in case of network congestion during playback.

One of the exciting features of PlayMail is the capability for a user to create his/her own face models with the option of making these face models available to the public. To this extent, the server acts as a hosting service for specialized contents. It provides for password-protected user access.

Fig. 24 shows the user interface for face model creation. Guided by the location of feature points on a cartoon image, the user marks the corresponding location of the feature point on the image that he submitted to the server. Using these correspondences, the server computes an interpolation function based on radial basis functions that deforms its face model template to match the new face [110]. That face model including its face animation tables is then ready to be animated using the face animation engine [111].



Place this point on the top of the eye at your left
You can already click on "Create" button if you want.

Fig. 24. The user marks feature points in the face manually in order to create a face model. Dots marked in the right, generic face model (see right upper eyelid) guide this process. [Reproduced with permission from 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. (Chichester, U.K.: Wiley, 2002) [61]].

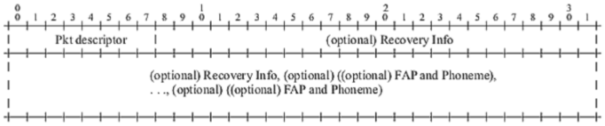

0 1 2 3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
Pkt descriptor
(optional) Recovery Info
(optional) Recovery Info, (optional) ((optional) FAP and Phoneme),
. . ., (optional) ((optional) FAP and Phoneme)

Fig. 25. PFAP payload (packet descriptor).

E. Streaming Protocols for Face Animation With Network-Based TTS Server

From a content provider perspective, the content presentation on the client terminal needs to be predictable. Therefore, the content provider must be able to define the face model as well as the voice of the talking face. Downloading a face model to the client is feasible today. However, the voice of a natural-sounding TTS is so large that it cannot be downloaded in a reasonable time in the foreseeable future. Therefore, content providers may favor an architecture where the face model is rendered locally on the client and the voice is computed on a network-based TTS server.

The interfaces shown in Fig. 2 are designed as real-time interfaces. In order to transport real-time data the real-time transport protocol (RTP) can be used. This protocol is based on an RTP header and a payload, which changes the format with the type of data in this payload. Different payload format definitions for audio and text data are specified, but not one for phonemes and FAPs. The following sections describing the Phoneme/FAP (PFAP) stream as RTP payload format [112].

The PFAP payload is designed to hold phonemes, FAPs, as well as recovery information for FAPs. These three types of data are transported using descriptors that specify the structure of each element. Fig. 25 shows the PFAP stream as an RTP payload format definition. Only the *Packet Descriptor* has a fixed position. All other descriptors are in the order of usage.

The *Packet Descriptor* has two functions: describe the type of recovery information (dynamical or complete) and define the type of descriptor (phoneme or FAP) following the optional recovery information.

The *Phoneme Descriptor* is designed to hold the phoneme symbol, phoneme duration, stress, and f0Average of this phoneme. The phoneme symbol is defined through a hexadecimal number, which leads together with a mapping table to the right phoneme out of a phoneme alphabet like DARPA or ARPAbet. The phoneme duration is given in units of milliseconds. Stress marks a stressed phoneme, and f0Average is the frequency of the synthesized audio signal for this phoneme in units of 2 Hz. The next descriptor (phoneme or FAP), end of packet, or end of text is defined using two information bits.

The *FAP Descriptor* defines a FAP, desired amplitude, transition, and a time curve according to the MPEG-4 specification. The valid range of FAPs transmitted via *FAP Descriptor* is defined from 3 to 74. Facial expressions start with FAP 69 (joy).

The PFAP RTP payload offers to recover FAP(s) in case packets get lost. Since the format is designed for real-time interactive services, we only recover state-like information, i.e., the facial expressions but not the transient phonemes. Dynamical recovery information may be transmitted with each regular packet. Complete recovery is transmitted as a separate packet. Dynamical recovery holds FAP(s) from previous n packets, and complete recovery recovers the state of the facial expressions and holds all FAP(s) with a nonzero amplitude.

The PFAP payload format enables using face animation with network-based TTS. The error resilient transport of phonemes and facial expressions requires a data rate of less than 800 b/s for text with many facial expressions. The latency of face animation using a network-based TTS server depends on the speed of the TTS server as well as on the payload format. While one PFAP packet may hold the phonemes and facial expressions of an entire sentence, the implementer may choose to pack a sentence into several shorter packets, thus reducing latency.

VIII. Conclusion

Lifelike talking faces for interactive services offer an exciting new modality for man–machine interactions. In this paper, we have covered the two principal ways talking heads can be implemented: model-based and sample-based. The model-based approach develops parametric models of shapes and movements. For real-time systems, the surface properties are not modeled using reflectance and transparency models but are implemented with texture maps. The shape and, hence, the texture map are deformed during animations, leading to artifacts in the rendered face that make it look less natural. The model-based heads can be built compactly, which makes them suited for real-time applications over low bandwidth connections.

The sample-based approach, on the other hand, opens the possibility of generating heads of such a quality that they may be mistaken for recordings of real humans. Such a model consists of a 3-D shape and a database of normalized and labeled mouth and eye images. For animations of a high quality, the database must contain tens of thousands of sample textures resulting in footprints that easily exceed 100 MB. It remains to be seen how much this size can be reduced without compromising the quality. We currently try to cover all appearances with recorded samples, thus incurring minimal or no deformation of the textures. A combination of sample-based and model-based techniques may integrate the best parts of the two approaches in the future.

Regardless of the model type, there are several essential elements that any face animation system must contain, including an audio and face renderer connected via a synchronization module, a coarticulation engine for creating the correct mouth shapes for the spoken text, a prosody analyzer, and a visual prosody synthesizer that creates the appropriate eye and head motion, based on the prosody of the spoken text. Up to now, most face animation system implemented visual prosody in an *ad hoc*, improvised manner. However, a carefully modeled visual prosody is essential for a natural appearance of an animation. Much more work remains to be done to analyze quantitatively prosodic movements of humans in various situations. Up to now, head movements have been analyzed either qualitatively or with motion capture equipment. However, the translation from the captured parameters to the animations of the head remains a problem, and little has been done to categorize prosodic movements, so that they could be added automatically to an animation.

Evaluation of face animation quality is critical for any progress. We developed metrics to determine the quality of a talking mouth based on criteria like mouth closures, mouth protrusions, turning points in lip motion direction, and motion smoothness. Subjective tests are still required to judge nonspeech-related animations like visual prosody or the effect of a face in human–computer interactions.

Finally, we presented several applications from our research. The News Reader and PlayMail applications can rely on streaming of video to the client. Interactive applications like Help Desk and E-cogent require low latency and rely on rendering the face model on the terminal of the user. We describe an RTP-based protocol that transports the required FAPs and phonemes from a server to a client in these applications. Since this protocol requires only data rates below 800 b/s in addition to the audio data, we can imagine the use of lifelike animated faces over telephone and wireless connections.

Currently, the demand on the rendering capabilities of a processor limits interactive dialogues with high-quality talking heads to PCs. The various schemes described in this paper illustrate how the talking head can be implemented in a stand-alone system or in a distributed manner via the Internet. With the rapidly increasing compute power found on PDAs and even cell phones, it is plausible that within the near future we will carry our friendly, personal assistant around with us and engage in a little chat when and wherever we feel like it.

For an interactive demonstration of our sample-based talking-head synthesizer, we encourage the reader to visit our Web site at http://vir2elle.com.

REFERENCES

[1] J. Gratch, J. Rickel, E. Andre, N. Badler, J. Cassell, and E. Petajan, “Creating interactive virtual humans: Some assembly required,” *IEEE Intell. Syst.*, vol. 17, pp. 54–63, July/Aug. 2002.

[2] A. Hunt and A. Black, “Unit selection in a concatenative speech synthesis system using a large speech database,” in *Proc. ICASSP*, vol. 1, 1996, pp. 373–376.

[3] J. Schroeter. The fundamentals of text-to-speech synthesis. presented at VoiceXML Forum. [Online] Available: http://www.voicexml.org/Review/Mar2001/features/tts.html

[4] W. L. Johnson, J. W. Rickel, and J. C. Lester, “Animated pedagogical agents: Face-to-face interaction in interactive learning environments,” *Int. J. Artif. Intell. Educ.*, vol. 11, pp. 47–78, 2000.

[5] C. Elliott, J. Rickel, and J. Lester, “Lifelike pedagogical agents and affective computing: An exploratory synthesis,” in *Lecture Notes in Computer Science, Artificial Intelligence Today*, M. Wooldridge and M. Veloso, Eds. Heidelberg, Germany: Springer-Verlag, 1999, vol. 1600, pp. 195–212.

[6] J. Lester, S. G. Towns, C. B. Callaway, J. L. Voerman, and J. FitzGerald, “Deictic and emotive communication in animated pedagogical agents,” in *Embodied Conversational Agents*, J. Cassell, J. Sullivan, S. Prevost, and E. Churchill, Eds. Boston, MA: MIT Press, 2000, pp. 123–154.

[7] E. Andre, T. Rist, and J. Muller, “Guiding the user through dynamically generated hypermedia presentations with a life-like character,” in *Proc. Intelligent User Interfaces*, 1998, pp. 21–28.

[8] T. Rist, E. Andre, and J. Muller, “Adding animated presentation agents to the interface,” in *Proc. Intelligent User Interfaces*, 1997, pp. 79–86.

[9] A. Don, T. Oren, and B. Laurel, “Guides 3.0,” in *Proc. Video ACM CHI*, 1993, pp. 447–448.

[10] S. Gibbs and C. Breiteneder, “Video widgets and video actors,” in *Proc. UIST*, 1993, pp. 179–185.

[11] A. E. Milewski and G. E. Blonder, “System and Method for Providing Structured Tours of Hypertext Files,” U.S. Patent 5 760 771, June 2, 1998.

[12] T. Bickmore, L. Cook, E. Churchill, and J. W. Sullivan, “Animated autonomous personal representatives,” in *Proc. Int. Conf. Autonomous Agents*, 1998, pp. 8–15.

[13] J. R. Suler, “From ASCII to holodecks: Psychology of an online multimedia community,” presented at the Convention Amer. Psychological Association, Chicago, IL, 1997.

[14] I. S. Pandzic, T. K. Capin, E. Lee, N. Magnenat-Thalmann, and D. Thalmann, “A flexible architecture for virtual humans in networked collaborative virtual environments,” in *Proc. Eurographics*, vol. 16, 1997, pp. 177–188.

[15] B. Damer, C. Kekenes, and T. Hoffman, “Inhabited digital spaces,” in *Proc. ACM CHI*, 1996, pp. 9–10.

[16] J. H. Walker, L. Sproull, and R. Subramani, “Using a human face in an interface,” in *Proc. ACM CHI*, 1994, pp. 85–91.

[17] L. Sproull, M. Subramani, S. Kiesler, J. H. Walker, and K. Waters, “When the interface is a face,” in *Proc. Human-Computer Interaction*, vol. 11, 1996, pp. 97–124.

[18] 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. Chichester, U.K.: Wiley, 2002, pp. 253–264.

[19] J. Ostermann and D. Millen, “Talking heads and synthetic speech: An architecture for supporting electronic commerce,” in *Proc. ICME*, 2000, p. MA2.3.

[20] S. Parise, S. Kiesler, L. Sproull, and K. Waters, “My partner is a real dog: Cooperation with social agents,” in *Proc CSCW*, 1996, pp. 399–408.

[21] Anthropics [Online]. Available: http://www.anthropics.com/

[22] Face2face animation [Online]. Available: http://www.f2fanimation.com

[23] Famous3D [Online]. Available: http://www.famous3d.com/

[24] LifeFX [Online]. Available: http://www.lifefx.com/

[25] Lipsinc [Online]. Available: http://www.lipsinc.com/

[26] Pulse3D [Online]. Available: http://www.pulse3d.com

[27] Y. Lee, D. Terzopoulos, and K. Waters, “Realistic modeling for facial animations,” in *Proc. ACM SIGGRAPH*, 1995, pp. 55–62.

[28] P. Kalra, A. Mangili, N. Magnenat-Thalmann, and D. Thalmann, “Simulation of facial muscle actions based on rational free form deformation,” in *Proc. Eurographics*, 1992, pp. 65–69.

[29] F. I. Parke and K. Waters, *Computer Facial Animation*. Wellesley, MA: A. K. Peters, 1996.

[30] I. S. Pandzic and R. Forchheimer, *MPEG4 Facial Animation—The Standard, Implementations and Applications*. Chichester, U.K.: Wiley, 2002.

[31] M. M. Cohen and D. W. Massaro, “Modeling coarticulation in synthetic visual speech,” in *Models and Techniques in Computer Animation*, N. Magnenat-Thalmann and D. Thalmann, Eds. Tokyo, Japan: Springer-Verlag, 1993, pp. 139–156.

[32] T. Ezzat and T. Poggio, “MikeTalk: A talking facial display based on morphing visemes,” *Proc. IEEE Computer Animation*, pp. 96–102, 1998.

[33] T. Ezzat, G. Geiger, and T. Poggio, “Trainable videorealistic speech animation,” in *Proc. ACM SIGGRAPH*, 2002, pp. 388–397.

[34] C. Bregler, M. Covell, and M. Slaney, “Video rewrite: Driving visual speech with audio,” in *Proc. ACM SIGGRAPH*, 1997, pp. 353–360.

[35] E. Cosatto and H. P. Graf, “Sample-based synthesis of photo-realistic talking heads,” *Proc. IEEE Computer Animation*, pp. 103–110, 1998.

[36] ——, “Photo-realistic talking heads from image samples,” *IEEE Trans. Multimedia*, vol. 2, pp. 152–163, Sept. 2000.

[37] *4020/RGB 3D Scanner With Color Digitizer*, Cyberware Laboratory, Inc., Monterey, CA, 1990.

[38] Eyetronics 3D scanning solutions [Online]. Available: http://www.eyetronics.com

[39] B. Guenter, C. Grimm, D. Wood, H. Malvar, and F. Pighin, “Making faces,” in *Proc. ACM SIGGRAPH*, 1998, pp. 55–66.

[40] F. Pighin, J. Hecker, D. Lichinski, R. Szeliski, and D. H. Salesin, “Synthesizing realistic facial expressions from photographs,” in *Proc. ACM SIGGRAPH*, 1998, pp. 75–84.

[41] V. Blanz and T. Vetter, “A morphable model for the synthesis of 3D faces,” in *Proc. ACM SIGGRAPH*, 1999, pp. 353–360.

[42] G. Kalberer and L. Van Gool, “Lip animation based on observed 3D speech dynamics,” in *Proc. SPIE*, vol. 4309, Videometrics and Optical Methods for 3D Shape Management, S. F. El-Hakim and A. Gruen, Eds., 2001, pp. 16–25.

[43] S. Kshirsagar and N. Magnenat-Thalmann, “Virtual humans personified,” in *Proc. Autonomous Agents Conf. (AAMAS)*, 2002, pp. 356–359.

[44] C. G. Fisher, “Confusions among visually perceived consonants,” *J. Speech Hearing Res.*, vol. 11, pp. 796–804, 1968.

[45] C. Pelachaud, “Visual text-to-Speech,” in *MPEG-4 Facial Animation: The Standard, Implementation and Applications*, I. S. Pandzic and R. Forchheimer, Eds. Chichester, U.K.: Wiley, 2002, pp. 125–140.

[46] *ISO/IEC IS 14496–2 Visual*, 1999.

[47] P. Ladefoged, *A Course in Phonetics*, 4th ed. Fort Worth, TX: Harcourt Brace Jovanovich, 2001.

[48] J. Beskow, “Rule-based visual speech synthesis,” in *Proc. Eurospeech*, 1995, pp. 299–302.

[49] I. Lehiste, *Suprasegmentals*. Cambridge, MA: MIT Press, 1970.

[50] N. Campbell and A. Black, “Prosody and the selection of source units for concatenative synthesis,” in *Progress in Speech Synthesis*, J. van Santen, R. Sproat, J. Olive, and J. Hirschberg, Eds. Berlin, Germany: Springer-Verlag, 1996, pp. 279–282.

[51] P. Ekman and W. V. Friesen, *Unmasking the Face. A Guide to Recognizing Emotions From Facial Clues*. Englewood Cliffs, NJ: Prentice-Hall, 1975.

[52] J. B. Bavelas and N. Chovil, “Faces in dialogue,” in *The Psychology of Facial Expression*, J. A. Russell and J. M. Fernez-Dos, Eds. Cambridge, U.K.: Cambridge Univ. Press, 1997, pp. 334–346.

[53] M. Argyle and M. Cook, *Gaze and Mutual Gaze*. Cambridge, U.K.: Cambridge Univ. Press, 1976.

[54] G. Collier, *Emotional Expression*. Hillsdale, NJ: Lawrence Erlbaum Assoc., 1985.

[55] M. Costa, T. Chen, and F. Lavagetto, “Visual prosody analysis for realistic motion synthesis of 3D head models,” in *Proc. Int. Conf. Augmented Virtual Environments and 3D Imaging*, 2001, pp. 343–346.

[56] H. P. Graf, E. Cosatto, V. Strom, and F. J. Huang, “Visual prosody: Facial movements accompanying speech,” in *Proc. Int. Conf. Automatic Face and Gesture Recognition*, 2002, pp. 396–401.

[57] K. Waters and T. Levergood, “An automatic lip-synchronization algorithm for synthetic faces,” in *Proc. ACM Multimedia*, 1994, pp. 149–156.

[58] J. Cassell, H. Vilhjálmsson, and T. Bickmore, “BEAT: The behavior expression animation toolkit,” in *Proc. ACM SIGGRAPH*, 2001, pp. 477–486.

[59] E. Vatikiotis-Bateson, K. G. Munhall, M. Hirayama, Y. Kasahara, and H. Yehia, “Physiology-based synthesis of audiovisual speech,” in *Proc. 4th Speech Production Seminar: Models and Data*, 1996, pp. 241–244.

[60] J. Ostermann, M. Beutnagel, A. Fischer, and Y. Wang, “Integration of talking heads and text-to-speech synthesizers for visual TTS,” presented at the ICSLP, Sydney, Australia, 1999.

[61] J. Ostermann, “PlayMail—Put words into other people’s mouth,” in *MPEG-4 Facial Animation: The Standard, Implementation and Applications*, I. S. Pandzic and R. Forchheimer, Eds. Chichester, U.K.: Wiley, 2002, pp. 241–251.

[62] R. Sproat and J. Olive, “An approach to text-to-speech synthesis,” in *Speech Coding and Synthesis*, W. B. Kleijn and K. K. Paliwal, Eds. Amsterdam, The Netherlands: Elsevier, 1995.

[63] J. Schroeter, A. Conkie, A. Syrdal, M. Beutnagel, M. Jilka, V. Strom, Y. J. Kim, H. G. Kang, and D. Kapilow, “A perspective on the next challenges for TTS research,” presented at the IEEE Signal Processing Workshop Speech Synthesis, Santa Monica, CA, 2002.

[64] M. Bulut, S. S. Narayanan, and A. K. Syrdal, “Expressive speech synthesis using a concatenative synthesizer,” presented at the ICSLP, Denver, CO, 2002.

[65] P. Hong, Z. Wen, and T. S. Huang, “Real-time speech-driven face animation with expressions using neural networks,” *IEEE Trans. Neural Networks*, vol. 13, pp. 100–111, Jan. 2002.

[66] M. Brand, “Voice puppetry,” in ACM SIGGRAPH, 1999, pp. 21–28.

[67] S. Nakamura, “HMM-based transmodal mapping from audio speech to talking faces,” in *Proc. IEEE Int. Workshop Neural Networks Signal Processing*, vol. 10, 2000, pp. 33–42.

[68] J. P. Lewis and F. I. Parke, “Automated lipsynch and speech synthesis for character animation,” in *Proc. ACM CHI+CG*, 1987, pp. 143–147.

[69] M. Nahas, H. Huitric, and M. Saintourens, “Animation of a B-spline figure,” *Vis. Comput.*, vol. 5, no. 3, pp. 272–276, 1988.

[70] X. Huang, F. Alleva, H. Hon, M. Hwang, K. Lee, and R. Rosenfeld, “The SPHINX-II speech recognition system: An overview,” *Comput. Speech Lang.*, vol. 2, pp. 137–148, 1993.

[71] L. Williams, “Performance driven facial animation,” in *Proc. ACM SIGGRAPH*, 1990, pp. 235–242.

[72] M. Tekalp and J. Ostermann, “Face and 2-D mesh animation in MPEG-4,” *Signal Process. Image Commun.*, vol. 15, pp. 387–421, 2000.

[73] F. Lavagetto and R. Pockaj, “The facial animation engine: Toward a high-level interface for the design of MPEG-4 compliant animated faces,” *Proc. IEEE CSVT*, vol. 9, no. 2, pp. 277–289, 1999.

[74] T. S. Perry, “And the Oscar goes to …,” *IEEE Spectr.*, vol. 38, pp. 42–49, 2001.

[75] P. S. Heckbert, “Survey of texture mapping,” *IEEE Comput. Graph. Appl.*, pp. 56–67, Nov. 1986.

[76] *ISO/IEC IS 14496–1 Systems*, 1999.

[77] *ISO/IEC 14 772–1: Information Technology—Computer Graphics and Image Processing—The Virtual Reality Modeling Language—Part 1: Functional Specification and UTF-8 Encoding*, 1997.

[78] J. Hartman and J. Wernecke, *The VRML Handbook*. Reading, MA: Addison-Wesley, 1996.

[79] *ISO/IEC IS 14496–3 Audio*, 1999.

[80] T. K. Capin, E. Petajan, and J. Ostermann, “Efficient modeling of virtual humans in MPEG-4,” in *Proc. ICME*, 2000, p. TPS9.1.

[81] ——, “Very low bitrate coding of virtual human animation in MPEG-4,” in *Proc. ICME*, 2000, p. TPS9.2.

[82] F. I. Parke, “Parameterized models for facial animation,” *IEEE Comput. Graph. Appl.*, vol. 2, pp. 61–68, Nov. 1982.

[83] K. Waters, “A muscle model of animating three dimensional facial expression,” *Comput. Graph.*, vol. 22, no. 4, pp. 17–24, 1987.

[84] J. Ostermann and E. Haratsch, “An animation definition interface: Rapid design of MPEG-4 compliant animated faces and bodies,” in *Proc. Int. Workshop Synthetic—Natural Hybrid Coding and Three Dimensional Imaging*, 1997, pp. 216–219.

[85] *Proc. 2nd Int. Workshop Recognition, Analysis and Tracking of Faces and Gestures in Realtime Systems*. Los Alamitos, CA: IEEE Comput. Soc. Press, 2001.

[86] *Proc. 5th Int. Conf. Automatic Face and Gesture Recognition*. Los Alamitos, CA: IEEE Comput. Soc. Press, 2002.

[87] H. P. Graf, E. Cosatto, and G. Potamianos, “Robust recognition of faces and facial features with a multi-modal system,” in *Proc. IEEE Conf. Systems, Man, and Cybernetics*, 1997, pp. 2034–2039.

[88] R. Y. Tsai, “A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses,” *IEEE J. Robot. Automat.*, vol. RA-3, pp. 323–344, Aug. 1987.

[89] D. G. Lowe, “Fitting parameterized three dimensional models to images,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, pp. 441–450, May 1991.

[90] D. Oberkampf, D. Dementhon, and L. Davis, “Iterative pose estimation using coplanar feature points,” Univ. Maryland, College Park, CVL, CAR-TR-677, 1993.

[91] D. F. Dementhon and L. S. Davis, “Model-based object pose in 25 lines of code,” *Int. J. Comput. Vis.*, vol. 15, pp. 123–141, 1995.

[92] H. Li, P. Roivainen, and R. Forcheimer, “3-D motion estimation in model-based facial image coding,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 545–55, June 1993.

[93] D. DeCarlo and D. Metaxas, “Optical flow constraints on deformable models with applications to face tracking,” *Int. J. Comput. Vis.*, vol. 38, no. 2, pp. 99–127, 2000.

[94] C. W. Wightman and D. Talkin, “The Aligner: Text to speech alignment using Markov models and a pronunciation dictionary,” presented at the 2nd ESCA/IEEE Workshop Speech Synthesis, New Paltz, NY, 1994.

[95] G. Potamianos and H. P. Graf, “Linear discriminant analysis for speechreading,” *Proc. IEEE Workshop Multimedia Signal Processing*, pp. 221–226, 1998.

[96] M. Beutnagel, A. Conkie, J. Schroeter, Y. Stylianou, and A. Syrdal, “The AT&T next-gen TTS system,” *J. Acoust. Soc. Amer.*, pt. 2, vol. 105, p. 1030, 1999.

[97] E. Cosatto, M.S. thesis, Ecole Polytechnique Federale Lausanne, Lausanne, Switzerland, 2002.

[98] K. Silverman, M. Beckman, J. Pitrelli, M. Ostendorf, C. Wightman, P. Price, J. Pierrehumbert, and J. Hirschberg, “TOBI: A standard for labeling English prosody,” in *Proc. ICSLP*, vol. 2, 1992, pp. 981–984.

[99] ITU Telecom, “Methodology for the subjective assessment of the quality of television pictures,”, Recommendation ITU-R BT.500–10 1, 2000.

[100] I. S. Pandzic, J. Ostermann, and D. Millen, “User evaluation: Synthetic talking faces for interactive services,” *Vis. Comput. J.*, vol. 15, no. 7–8, pp. 330–340, 1999.

[101] J. J. Williams and A. K. Katsaggelos, “An HMM-based speech-to-video synthesizer,” *IEEE Trans. Neural Networks (Special Issue on Intelligent Multimedia)*, vol. 13, pp. 900–915, July 2002.

[102] D. W. Massaro, *Perceiving Talking Faces: From Speech Perception to a Behavioral Principle*. Cambridge, MA: MIT Press, 1998.

[103] E. Levin, R. Pieraccini, W. Eckert, G. DiFabbrizio, and S. Narayanan, “Spoken language dialogue: From theory to practice,” presented at the IEEE Workshop Automatic Speech Recognition and Understanding (ASRU), Keystone, CO, 1999.

[104] A. L. Gorin, J. H. Wright, G. Riccardi, A. Abella, and T. Alonso, “Semantic information processing of spoken language,” presented at the ATR Workshop MultiLingual Speech Communication, Kyoto, Japan, 2000.

[105] A. L. Gorin, J. H. Wright, and G. Riccardi, “How may I help you?,” *Speech Commun.*, vol. 23, pp. 113–127, 1997.

[106] (2001) SMIL, Synchronized Multimedia Integration Language (SMIL 2.0) Specification. W3C Proposed Recommendation. [Online]. Available: http://www.w3.org/TR/smil20/

[107] PlayMail [Online]. Available: http://playmail.research.att.com

[108] Real Networks [Online]. Available: http://www.real.com

[109] Microsoft [Online]. Available: http://www.microsoft.com

[110] A. C. Andres del Valle and J. Ostermann, “3D talking head customization by adapting a generic model to one uncalibrated picture,” in *Proc. ISCAS 2001*, vol. 2, 2001, pp. 325–328.

[111] M. V. Mani and J. Ostermann, “Cloning of MPEG-4 face models,” presented at the Int. Workshop Very Low Bitrate Video Coding (VLBV 01), Athens, Greece, 2001.

[112] J. Ostermann, J. Rurainsky, and R. Civanlar, “Real-time streaming for the animation of talking faces in multiuser environments,” presented at the ISCAS 2002, Phoeniz, AZ, 2002.

Eric Cosatto (Member, IEEE) received the M.S. degree in computer science and the Ph.D. degree in engineering science from the Swiss Federal Institute of Technology, Lausanne, Switzerland, in 1991 and 2002, respectively.

He is a Principal Member of Technical Staff at AT&T Labs–Research, Middletown, NJ. He has authored or coauthored over 30 papers and holds six patents with several more pending. His research interests include computer graphics, computer user interfaces, image understanding, and image processing.

Jörn Ostermann (Senior Member, IEEE) received the Dr.-Ing. degree from the University of Hannover, Hannover, Germany, in 1994.

From 1988 to 1994, he was a Research Assistant at the Institut für Theoretische Nachrichtentechnik, Hannover, Germany, conducting research in object-based analysis-synthesis video coding. In 1994, he joined the Visual Communications Research department at AT&T Bell Labs, Holmdel, NJ. He has been with AT&T Labs–Research, Middletown, NJ, since 1996. He has contributed to more than 50 papers, book chapters, and patents. He is Coauthor of the textbook *Video Processing and Communications* (Englewood Cliffs, NJ: Prentice-Hall, 2001). His current research interests include multimodal human–computer interfaces, talking avatars, and streaming.

Dr. Ostermann was a scholar of the German National Foundation. In 1998, he received the AT&T Standards Recognition Award and the ISO Award. He chaired the European COST 211 sim group coordinating research in object-based video coding and the Adhoc Group on Coding of Arbitrarily Shaped Objects in MPEG-4 Video. He is a Member of the IEEE Technical Committee on Multimedia Signal Processing, a Past Chair of the IEEE CAS Visual Signal Processing and Communications Technical Committee, and a Distinguished Lecturer of the IEEE CAS Society.

Hans Peter Graf (Fellow, IEEE) received the Diploma and Ph.D. degrees in physics from the Swiss Federal Institute of Technology, Zurich, Switzerland, in 1976 and 1981, respectively.

Since 1983, he has been with AT&T—first with Bell Laboratories, and since 1996 with AT&T Labs—working on neural net models, designing microelectronic processors, and building vision systems for industrial applications. He is currently a Technology Consultant at AT&T Labs–Research, Middletown, NJ, leading research projects on visual text-to-speech for interactive Web services. Massively parallel processors of his design were key components in high-speed address readers and document analysis systems. He developed image analysis algorithms for finding faces in video sequences and for recognizing facial parts under difficult conditions. This work led to the start of a sample-based computer graphics project that eventually evolved into the design of video-realistic talking heads. He has published over 100 articles and has some 25 patents issued or pending. His current research interests include graphics, recognition, and animation techniques that can make talking heads look more lifelike.

Dr. Graf is a Member of the American Physical Society.

Juergen Schroeter (Fellow, IEEE) received the Dipl.-Ing. and Dr.-Ing. degrees in electrical engineering from Ruhr-Universitaet Bochum, Bochum, Germany in 1976 and 1983, respectively.; AT&T Bell Laboratories, 1986–1995, AT&T Labs–Research, 1996-present.

From 1976 to 1985, he was with the Institute for Communication Acoustics, Ruhr-Universitaet Bochum, Bochum, Germany, where he did research in the areas of hearing and acoustic signal processing. Since 1986, he has been with AT&T Bell Labs, where he has been working on speech coding and synthesis methods employing models of the vocal tract and vocal cords. He is currently a Division Manager with AT&T Labs–Research, where he is leading efforts in visual text-to-speech, as well as leading the team that created AT&T's Next-Generation Text-to-Speech synthesis system. He is a Former Associate Editor of the *Journal of the Acoustical Society of America*.

Dr. Schroeter is a Fellow of the Acoustical Society of America. In 2001, he received the AT&T Science and Technology Medal. He is currently an Associate Editor for IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING.