Photo-Realistic Talking-Heads from Image Samples

Eric Cosatto and Hans Peter Graf, *Fellow, IEEE*

***Abstract*—This paper describes a system for creating a photo-realistic model of the human head that can be animated and lip-synched from phonetic transcripts of text. Combined with a state-of-the-art text-to-speech synthesizer (TTS), it generates video animations of talking heads that closely resemble real people. To obtain a naturally looking head, we choose a "data-driven" approach. We record a talking person and apply image recognition to extract automatically bitmaps of facial parts. These bitmaps are normalized and parameterized before being entered into a database. For synthesis, the TTS provides the audio track, as well as the phonetic transcript from which trajectories in the space of parameterized bitmaps are computed for all facial parts. Sampling these trajectories and retrieving the corresponding bitmaps from the database produces animated facial parts. These facial parts are then projected and blended onto an image of the whole head using its pose information. This talking head model can produce new, never recorded speech of the person who was originally recorded. Talking-head animations of this type are useful as a front-end for agents and avatars in multimedia applications such as virtual operators, virtual announcers, help desks, educational, and expert systems.**

***Index Terms*—Facial animation, talking-heads, sample-based image synthesis, face recognition, computer vision. .**

I. Introduction

ANIMATED characters, and talking heads in particular, are playing an increasingly important role in computer interfaces. An animated talking head attracts immediately the attention of a user, can make a task more engaging and adds entertainment value to an application. Seeing a face makes many people feel more comfortable interacting with a computer. For learning tasks several researchers report that animated characters can increase the attention span of the user, and hence improve learning results. When used as avatars, lively talking heads can make an encounter in a virtual world more engaging. Today such heads are usually either recorded video clips of real people or cartoon characters lip-synching synthesized text.

Often a cartoon character or robot-like face may do, yet we respond to nothing as strongly as to a real face. For an educational program, for example, a real face is preferable. A cartoon face is associated with entertainment, not to be taken too seriously. An animated face of a competent teacher, on the other hand, can create an atmosphere conducive to learning and therefore increase the impact of educational software.

Generating animated talking heads that look like real people is a very challenging task, and so far all synthesized heads are still far from reaching this goal. To be considered natural, a face has to be not just photo-realistic in appearance, but it must also exhibit proper plastic deformations of the lips synchronized with the speech, realistic head movements and emotional expressions. We are trained since birth to recognize faces, facial expressions and speech postures and therefore are highly sensitive to the slightest imperfections in a synthesized talking face.

To synthesize talking-head animations, our system proceeds in two steps. In the first step, video footage of a person talking is analyzed to extract image samples of several facial parts, such as mouth, eyes, forehead etc. This step is typically performed only once, off-line, and results in a database of facial parts samples. In the second step we use a phonetic transcript from a text-to-speech synthesizer (TTS) to select and reassemble facial part samples into an animation. This synthesis is performed on-line, for each animation. We create a model of the head of the recorded person in such a way that it can be used for both the analysis and the synthesis steps. Our head model comprises two main parts: a rough three-dimensional (3-D) polygon model and a set of textures for each polygon. Three-dimensional polygons marking the position of facial parts on the subject's head are used to model the rigid movements of the head. Their relative 3-D positions are recorded from the subject's head. These polygons do not capture the fine details of the shapes of facial parts. Instead, each facial part is modeled by a set of captured image samples representative of its appearances. This sample-based modeling approach preserves a high level of detail in the appearance of the face.

This paper is organized as follows. Section II compares our approach to other work, Section III defines how the head and its facial parts are modeled, Section IV presents the process of capturing and analyzing video data and generating a database of image samples, and Section V describes the synthesis of the talking-head animation from a phonetic transcript.

II. Previous Work

The human face has been a topic of high interest in both the computer graphics community [16], [35], as well as the pattern recognition community since a long time. In computer graphics, researchers have strived toward realism in animated head models, while pattern-recognition researchers have been dealing with recognizing human faces in images. Our system draws on results and techniques from both of these communities. Combining machine vision with computer graphics is an idea that is receiving increasing attention recently [5], [10], [14], [20].

In computer graphics, many different techniques exist for modeling the human head, achieving various degrees of realism and flexibility. Most approaches use 3-D meshes to model in fine detail the shape of the head [17], [31], [33]. These models are usually created using advanced 3-D scanning techniques,

Manuscript received April 27, 1999; revised June 16, 2000. The associate editor coordinating the review of this paper and approving it for publication was Prof. Ulrich Neumann.

The authors are with the AT&T Labs-Research, Red Bank, NJ 07701-7033 USA (e-mail: eric@research.att.com; hpg@research.att.com).

Publisher Item Identifier S 1520-9210(00)07586-6.

such as a CyberWare® range scanner. Some of them include information on how to move vertices according to physical properties of the skin and the underlying muscles [31]. Another approach is to fit a generic 3-D model to image data by solving a dynamic system incorporating either optical flow and edge constraints [24] or stereo disparity maps [36]. A technique for matching a 3-D model to a single photograph is proposed in [5], where the 3-D model is deformed in the directions of several principal components to fit the given two–dimensional (2-D) image. To obtain a natural appearance, these 3-D models typically use a 2-D image of the subject that is texture-mapped onto the 3-D model. As long as these 3-D models do not incur any plastic deformations, but only rigid movements of the entire head, images of high quality can be produced (providing the 3-D shape has been captured with high precision). However, when plastic deformations occur, the shape of some polygons of the 3-D model will change while the pixels from the 2-D image that have been allocated to these polygons will remain the same, resulting in "stretch" and "squeeze" artifacts. While this is tolerable for small deformations of surfaces with little texture, the animation of a human mouth articulating speech, because of the hundreds of little wrinkles on the lips and the texture of the surrounding skin, results in unnatural appearances and a "synthetic look."

An alternative approach is based on morphing between 2-D images. These techniques can produce photo-realistic images of new shapes by interpolating between two existing shapes. Morphing of a face requires precise specifications of the displacements of many points in order to guarantee that the results look like real faces. Most techniques, therefore, rely on a manual specification of the morph parameters [26]. Beymer *et al.* [25] and Bichsel [27] have proposed image analysis methods where the morph parameters are determined automatically, based on optical flow. While this approach gives an elegant solution to generating new views from a set of reference images, one still has to find the proper reference images. Moreover, since the techniques are based on 2-D images, the range of expressions and movements they can produce is rather limited. Ezzat *et al.* [10] have demonstrated a sample-based talking head system that uses morphing to generate intermediate appearances of mouth shapes from a very small set of manually selected mouth samples. While morphing generates smooth transitions between mouth samples, this system does not model the whole head and does not synthesize head movements and facial expressions.

Recently, there has been a surge of interest in sample-based techniques (also referred to as data-driven) for synthesizing photo-realistic scenes. These techniques generally start by observing and collecting samples that are representative of the signal to be modeled. The samples are then parameterized so that they can be recalled at synthesis time. Typically, samples are processed as little as possible to avoid distortions. A talking-head synthesis technique based on recorded samples that are selected automatically has been proposed in [20]. This system can produce videos of real people uttering text they never actually said. It uses video snippets of tri-phones (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.

Other researchers explored ways of sampling both texture and 3-D geometry of faces [8], [9], producing realistic animations of facial expressions. These systems use multiple cameras or facial markers to capture the 3-D geometry and texture of the face in each frame of video recordings. Deriving the exact geometry in the mouth and mouth cavity areas as they undergo plastic deformations remains difficult, however. Extensive manual measuring of the images in [9] and a complex data capture setup involving six cameras and hundreds of facial markers in [8], are required resulting in a labor intensive capture process, especially when tens of thousand of video frames have to be analyzed, making them unsuitable for speech production.

In the area of pattern recognition, many systems have been described that locate and recognize facial features on the human head [2]. Most facial recognition systems focus on recognizing the identity of persons on an image [34], recognizing facial expressions [19], [29] or tracking the pose of heads on images [1]. Most of these systems are tuned for locating facial parts with "reasonable" accuracy (within about five pixels) on a large database of different subjects. To be able to synthesize a photo-realistic talking-head, we need to locate the position of facial parts with subpixel accuracy. Misalignment of even one pixel (of a bitmap of 150×100 pixels) results in visible jerkiness during the animation. These artifacts are particularly annoying because the facial parts appear to float over the base face, ruining the realism of the entire animation. Our facial locator module is trained on a particular person and proceeds in several stages with increasing accuracy in order to locate facial parts with sufficient precision.

Other systems are good at tracking with high accuracy features of the face as it moves [7], [18], but often require some manual bootstrapping. In order to analyze tens of thousands of video frames segmented in hundreds of separate video files, we found these methods inadequate for our application because it would require too often a manual intervention. A bottom-up, model-based system to locate individual facial parts has been proposed by Saulnier *et al.* [30]. This system shares many aspects of the first analysis phase of our approach. However, we refine measurements obtained by the first-phase using matched filter techniques. An earlier version of the present system has been described in [11].

III. Head Model

A key problem with sample-based techniques is to control the number of image samples that need to be recorded and stored. A face's appearance changes due to talking, emotional expressions, and head orientation, leading to a combinatorial explosion in the number of different appearances. To keep the number of samples at a manageable level, we divide the face into a hierarchy of parts and model each part separately. To factor out rigid movements of the entire head, we use a simple 3-D model and derive the head pose by matching the 3-D model to corresponding points on the sample image. This results in a compact

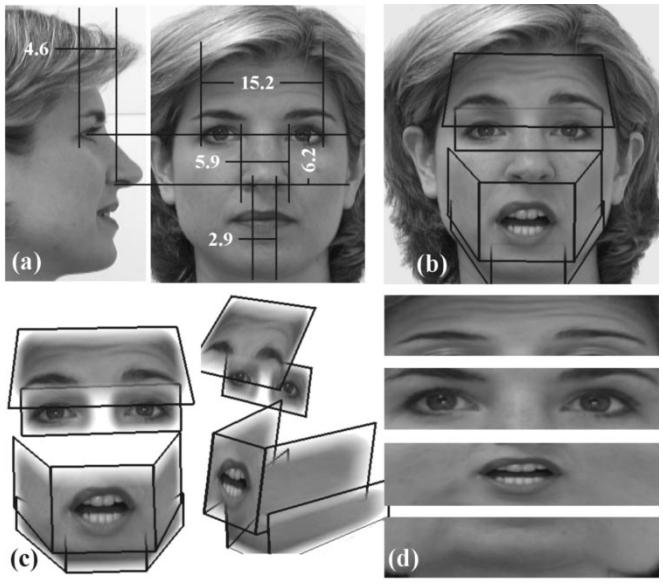

Fig. 1. Head model. (a) Measurements from the subject's head provide the 3-D location of facial points. (b) From these points, planar quads are obtained that mark areas of facial parts. (c) The pose of the head determines the position of the projection of these quads on an image. (d) The pixels bound by these projected quads are extracted into normalized bitmaps.

model that can create animations with head movements, speech articulation, and different emotional expressions, and combines the flexibility of 3-D models with the realism of images. Our face model is defined as follows.

1) Hierarchy of Parts: The head is separated into a "base face" [Fig. 1(b)] and a number of facial parts. The base face covers the area of the whole face serving as a substrate onto which the facial parts are integrated. The facial parts are: mouth with cheeks and jaw, eyes, and forehead with eyebrows [Fig. 1(d)]. Nose and ears are not modeled separately, but are part of the base face. There is no unique way of decomposing a face into parts, and no part of the face is truly independent from the rest. Muscles and skin are highly elastic and tend to spread a deformation in one place across a large part of the whole face. Nevertheless, there exists a large degree of independence between these facial parts. The decomposition described here was chosen after studying how facial expressions are generated by humans [42] and how they are depicted by artists and cartoonists [41].

2) Three-Dimensional Head Model: The 3-D shape of each facial part is approximated with a small number of 3-D four-sided polygons (or planar quads) [Fig. 1(c)]. These quads are derived from measurements obtained from the subject's head. Using the pose of the head, the 3-D positions of these quads are calculated and their perspective projection onto the image is obtained [Fig. 1(b)]. During the analysis phase, pixels bound by the projected quads are "un-warped" into bitmaps which are now corrected for the head orientation [Fig. 1(d)]. During the synthesis phase, these normalized bitmaps are warped into areas bound by the projected quads. Hence, the same 3-D head model is used for analysis (creating the database of facial parts samples from video data) and synthesis (generating new facial appearances by combining facial parts samples).

3) Sample Bitmaps of Facial Parts: For each facial part, sample bitmaps are recorded that cover the range of possible appearances produced by plastic deformations. For example, for speech production, sample images of the mouth uttering speech are recorded. Using the 3-D model of the facial parts, the sample bitmaps are extracted from the original images, factoring out the head pose, and, thus, no separate bitmaps have to be recorded to account for different head orientations. We currently limit ourselves to frontal views of the head, allowing rotations of the skull that are typical during the production of spontaneous speech. By studying movements of the head of five subjects over about 1 h of speech, we found that the range of movements does not exceed 10° for the yaw, 7.5° for the pitch and 4.8° for the roll (see Table I). This range of rotation does not introduce significant deformations or occlusions of the prominent facial parts (eyes, mouth, chin, forehead), making it possible to model the 3-D shape of the head with such a small number of planar quads; a fact noticed earlier by [29]. Finally, each sample bitmap is labeled with a set of features, so that it can be retrieved efficiently from the database at synthesis time. The type and number of features used depend on the facial part and the application and are further discussed in Section IV-G.

4) Sample Bitmaps of the Base Face: Some parts of the head are not incurring significant plastic deformations during the production of speech and emotions. These parts include the nose, the ears, and the hair. However, these facial parts often become partly occluded when the head is moving, making it difficult to use warping to model these movements. Instead of trying to model the base face with its 3-D shape, we simply keep the recorded image of the entire head along with the pose information. At synthesis time, for a given base head image, sample bitmaps of facial parts are integrated onto such base heads using the pose information (see Section V-E).

IV. Analysis

A. Data Capture

First, measurements are made on the subject's face to determine its geometry [Fig. 1(a)]. From two calibrated front and profile images of the subject, we measure the relative distances of the following facial points: the four eye corners, the two nostrils, the bottom of the upper front teeth, the chin, and the base of the ears. Quads marking the mouth-cheeks-chin areas, eyes area, and forehead area are derived from these initial measurements. Since this is done only once, and only ten points are involved, there is little incentive to automate it. Techniques exist, such as the ones described in [17], [24], and [36], that can adapt a generic head model to a specific person from video sequences showing head movements. This may be useful if only video footage exists without the person being present.

A person is then recorded while speaking freely in front of the camera. For the examples shown here, the lady spoke 14 phrases, each 2 to 3 s long. We try to keep the capture process as simple and nonintrusive as possible, since we are interested in capturing the typical head movements during speech as well as unique ways of articulating words. In particular, we avoid

TABLE I
FOR SIX DIFFERENT SUBJECTS RECORDED WHILE SPEAKING FACING THE CAMERA (A TOTAL OF OVER 200 000 FRAMES), WE ANALYZED THE RANGE OF EACH ANGLE OF THE HEAD POSE: THE ROTATION AROUND X AXIS (YAW), THE Y AXIS (PITCH) AND THE Z AXIS (ROLL). THE RESULTS SHOW AN AVERAGE RANGE OF 10.4° FOR THE YAW, 7.5° FOR THE PITCH, AND 4.8° FOR THE ROLL

subj	set	nb frames	x min	x avg	x max	x Rng	y min	y avg	y max	y rng	z min	z avg	z max	z rng
tracie2	s1	83094	-8.6	-6.6	-4.5	4.1	-2.7	-1.5	0.1	2.6	1.3	2.2	3.7	2.4
tracie2	s2	20122	-12	-6.3	-4	8	-3.5	0.02	1.35	4.8	1.2	2.2	3.3	2.1
kris	s1	28178	-6.7	0.4	6	12.7	-10	-4.3	2.3	12.3	-5.4	-1.4	2.2	7.6
dawn	s1	25664	-11.7	-8.7	-5.2	6.5	-2.9	-1	0.9	3.8	-1.3	0.7	4.8	6.1
tracie	s1	23494	-5	-2.4	0.4	4.6	-5	-1.1	3	8	-2.6	-0.6	2.3	4.9
Sheryl	s1	27648	-15	-9.2	11.4	26.4	-8	-1.4	5.8	13.8	-4	0.2	2	6
avg. range						10.4				7.5				4.8

any head restraints or forced pose, such as requiring the subject to watch constantly in a given direction. Thanks to robust face location recognition algorithms, we do not need any special markers on the subject's face, but rather exploit the natural richness of features of the face. We also avoid the need of multiple cameras. Knowing the positions of a few points in the face allows to recover the 3-D head pose from 2-D images, using techniques described in Section IV-E. Lip movements can be extremely fast, which may cause blurry images when the frame rate is not high enough. We therefore record 60 fields/s instead of 30 frames. Artificial lights illuminate the subject homogeneously so that head movements do not change the luminance within facial parts too much. We also record the subject in one session, in order to make sure illumination conditions do not change. Moreover, having a background of uniform and neutral color makes finding the location of the head easy. Our setup consists of one camera, a teleprompter, two 300 W lights and a background panel. The subject reads text from the teleprompter ensuring a clear frontal view of the face [Fig. 5(a)]. The audio and video is recorded uncompressed on a digital D-1 tape and later compressed into MPEG2 video files and PCM audio files. We capture frames of 560 $\times$ 480 pixels in size with the head being about four-fifths of this height, resulting in a high level of details of facial features, skin and hair.

Quite some effort has gone into developing the whole system in a way that the capturing process remains easy and cheap. Eventually, the system should be usable outside the lab by relatively unskilled personnel, or even at home by the user himself. The final goal is to have an easy procedure where you can quickly produce a talking head model of yourself.

B. Locating the Face

Sample-based synthesis of talking heads depends on a reliable and accurate recognition of the face and of the positions of facial features. Without an automatic technique for extracting and normalizing facial features, a manual segmentation of the images has to be done. Considering that we need samples of all lip shapes, of different head orientations, and of several emotional expressions, thousands of images have to be searched for the proper shapes. If we also want to analyze the lip movements during transitions between phonemes, we have to analyze hundreds of thousands of images. Clearly, it is not feasible to do such a task manually.

The main challenge for the face recognition system is the high precision with which the facial features have to be located. An error as small as a single pixel in the position of a feature distorts the pose estimation of the head noticeably. Moreover, when parts of a face are integrated into a new face, they have to be placed with an accuracy of one pixel or better. Otherwise, the animations look jerky and lose their natural appearance. In order to achieve such a high precision, our analysis proceeds in three steps, each one with an increased accuracy. The first step finds a coarse outline of the head plus estimates of the positions of the major facial features. In the second step, the areas around the mouth, the nostrils, and the eyes are analyzed in more detail. The third step, finally, zooms in on specific points of facial features, such as the corners of the eyes, of the mouth, and of the eyebrows, and measures their positions with high accuracy.

In a first step, the whole image is searched for the presence of heads and the locations of the main facial features. This is rather easy, since there is typically only one face present in the image, which is placed in the center of the image, and the illumination is carefully controlled. We summarize briefly the steps of this analysis here. Further details can be found in [3] and [21].

Each frame is analyzed with two different algorithms. Color segmentation and texture segmentation. For the texture segmentation, we use the luminance of the image. First, a bandpass filter removes the highest and lowest spatial frequencies [Fig. 2(a)]. Then a morphological operation, followed by adaptive thresholding, produces a binary image, where the black blobs mark areas with spatial frequencies and sizes typical of facial features such as eyes and mouth [Fig. 2(b)]. After a connected component analysis, the center of mass, width, and height are computed for each blob of connected pixels.

For the color segmentation, the hue is divided into a range of background colors and a range of skin or hair colors. These ranges are obtained initially by manually sampling a few labeled images from the recordings. After thresholding and connected component analysis, the same features are extracted from the blobs as in the texture analysis.

Combinations of these features are evaluated in a bottom-up approach in order to locate eyes, eyebrows, nostrils, mouth, and the outline of the head. First, each connected component is tested whether it may represent a facial feature. A 2-D head model is used to provide acceptable ranges for sizes and relative positions of facial features within the face. This model

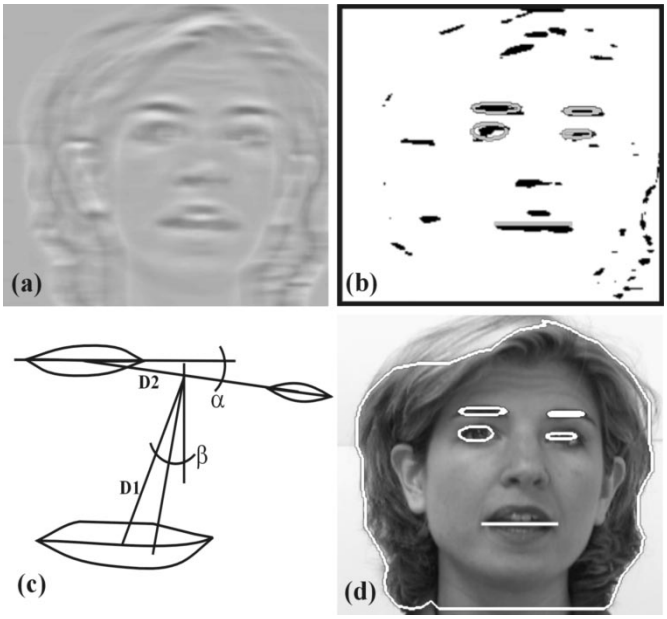

Fig. 2. Recognition process. The image is filtered with (a) a bandpass filter followed by a morphological operation and (b) adaptive thresholding to generate a binary image. Combinations of blobs are evaluated whether they might represent facial features. (c) Features used for identifying blobs as eye pair and mouth: $d = D2/D1$ and $\delta = |\alpha - \beta|$. (d) The combination of blobs matching closest the reference values of the model are taken as locations of the facial features.

was initially obtained by averaging facial feature sizes and positions across five subjects and is further refined by sampling a few labeled images of the subject itself. Checking width and height against this model, a connected component may be accepted as candidate for an eye, eyebrow, nostril, or mouth. These candidates are given a score based on their difference in size from that of the model, while the rest of the connected components are discarded. In the next step, combinations of two facial part candidates are evaluated. For example, all the candidates marked as eyes are checked against the model, whether any two of them could represent an eye pair. Then more complex combinations of connected components are tested for eyes—eyebrows, eyes—nostrils, and eyes—mouth. Fig. 2(c) illustrates the features used for identifying an eye pair plus mouth combination. Each combination is given a score based on the difference in relative distances between the candidate facial parts and the corresponding facial parts in the model. The score obtained in the previous step is also propagated and a combined score is obtained. In the final step, all combinations representing a whole head are evaluated (whole head = eye-pair + eye-brow pair + nostril pair + mouth) and the highest scoring one is retained. Missing features do not disqualify combinations (e.g., a whole head combination might exist where no eyebrows were found), instead no score is added for the missing feature, resulting in a lower score for the combination. This bottom-up approach produces reliably and quickly the location of the head as well as the positions of the major facial features [Fig. 2(d)]. A typical recognition accuracy obtained on a set of 9700 frames is as follows: eyes and mouth—97.5% and eye brows—42% (in this case the hair were often hanging over the eye brows).

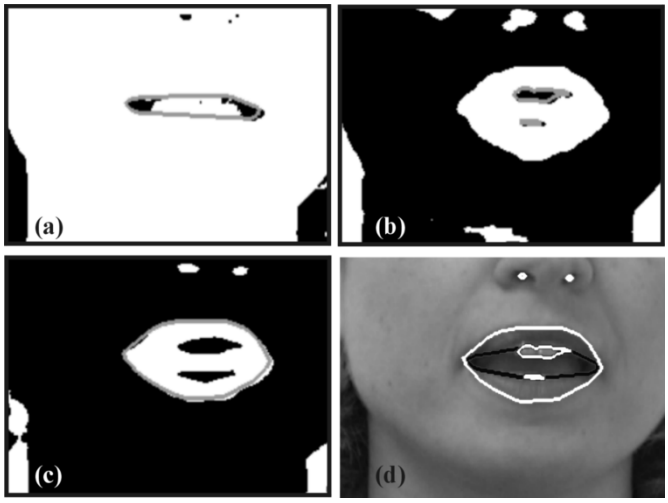

Fig. 3. Color segmentation of the area around the mouth identifies dark areas inside (a) the mouth, (b) the teeth, and (c) the outlines of the lips. (d) The blobs in the binary images are tested for their sizes and relative positions in order to identify the parts of the mouth.

C. Measuring Facial Features

Finding the exact dimensions of the facial features is more challenging, since the person is moving the head during the recordings and is changing facial expressions while speaking. This leads to variations in the appearance of a facial feature and locally affects the lighting conditions. For example, during a nod, a shadow may fall over the eyes. The analysis described earlier produces an approximate measure of the positions of facial features, but not an accurate description of their shapes. We therefore need to further analyze the areas around eyes, mouth, and the lower end of the nose.

The algorithm proceeds with color segmentation of the areas around the mouth and the eyes. In a training phase, using a few sample images, these areas were analyzed manually. A leader-clustering of the hue-saturation-luminance color space produced patches (blobs of similar color) that were labeled to identify various parts of the mouth and the eyes. Now, an image of the mouth area, for example, is thresholded for each of the color clusters, followed by a connected component analysis on each of the resulting binary images [see Fig. 3(a)–(c)]. By analyzing the shapes of the connected components and their relative positions in the way described in Section IV-B, we assign the colors to the teeth, the lips, and the inner, dark part of the mouth. This straightforward color segmentation typically suffices, since we deal with videos taken under well-controlled lighting conditions.

fThe accuracies achieved for the dimensions of the mouth are typically ± 2 pixels (standard deviation), where the width of the mouth is 60–100 pixels. However, sometimes, for example when the speaker smiles, strong shadows cover the mouth corners, leading to larger errors in the measurements of the mouth width. Moreover, color segmentation may not work well if the lip color is not sufficiently different from that of the surrounding skin. This is sometimes the case for elderly people or male speakers with narrow lips or with facial hair. We tested the accuracy of lip measurements in a large experiment for automatic lip reading [15], where 50 different people pronouncing over

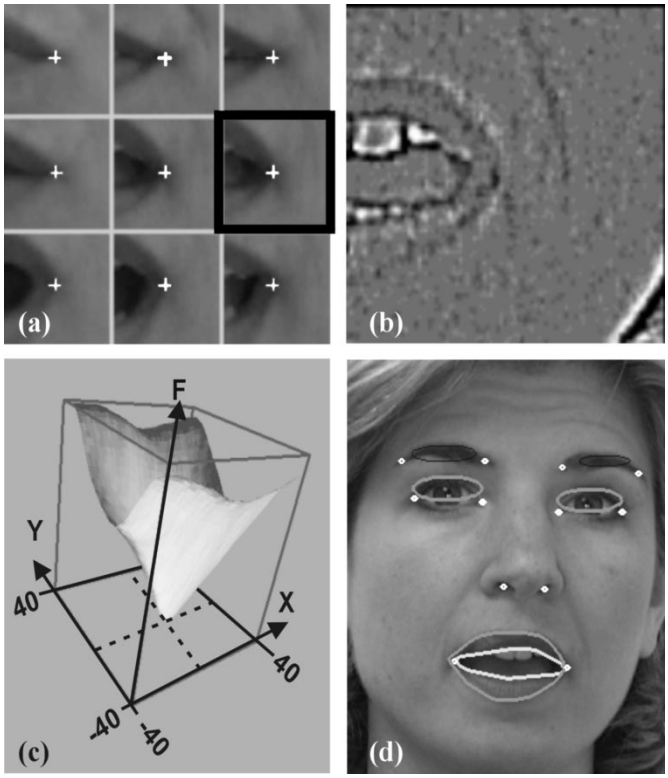

Fig. 4. Correlation analysis selects a kernel representing (a) a mouth corner and scans it over the area of the right half of the mouth. (b) The image as well as the kernel is filtered before the correlation. (c) The correlation function shows a prominent minimum at the position of the mouth corner. (d) This analysis provides the exact locations of eye corners, brow corners, mouth corners and nostrils.

5000 utterances were recorded under varying lighting conditions. While color segmentation very often provides the lip outlines with remarkable accuracy, its robustness is not always sufficient. Recalibrating the color thresholds dynamically during the analysis by applying color clustering every 20–50 frames can track changes in lighting conditions and improves the results. Combining the color segmentation with texture segmentation, as described in Section IV-B, can also improve robustness. Yet, after experimentation with these approaches, we developed an alternative method, which is more robust and finds feature points with a higher accuracy.

D. High Accuracy Feature Points

Correlation analysis is well suited to determine the location of feature points in the face with the precision needed for measuring the head pose. In a fully automatic training phase, representative examples of feature points, such as eye-corners, nostrils, and mouth-corners, are selected from the recorded video, using geometric features (Section IV-C) and the area around these points is cut out, generating, for each feature point, a set of kernels. For example, for the left mouth corner, nine examples are selected [Fig. 4(a)] based on the width and height of the mouth (three different widths $\times$ three different heights). During the analysis phase, for an image and a given feature point, a kernel is chosen with dimensions that are most similar to the ones on the image, and this kernel is scanned over the area around the feature point.

In order to verify that such an approach works reliably, we studied the shapes of the correlation functions around the feature points. A correlation analysis or matched filter approach works well if the kernel resembles closely the image to be located. Feature points, such as mouth corners, change significantly in appearance depending on the head orientation, the lighting conditions, and whether the mouth is open or closed. Therefore, a single kernel will not be sufficient for an accurate determination of the position. Tests with different mouth images indicated that nine kernels cover the range of appearances encountered here [see Fig. 4(b)]. The correlation function shows a prominent global minimum at the location of the mouth corner, as shown in Fig. 4(c). Moreover, the correlation function is monotonically decreasing toward the minimum. Qualitatively, the same behavior is observed for all the feature points we investigated, which makes the correlation analysis suited for a gradient descent approach, saving considerable computation time. Instead of a full correlation analysis, where the kernel is scanned point-by-point over the whole image, we use a conjugate gradient technique [23], which finds the minimum typically in about ten steps. It requires numerically evaluating the gradient of the correlation function at each step. Yet, even with this additional computation, the conjugate gradient technique is one to two orders of magnitude faster than the full correlation. A typical size of the kernels is 20 $\times$ 20 pixels searching an area of 100 $\times$ 100 pixels, resulting, for ten iterations, in a computation time of less than 20 ms on a 300 MHz PC.

Only the luminance or the hue is used for this analysis and the kernel, as well as the image, is filtered with a high-pass filter before the correlation [Fig. 4(b)]. For each mouth corner, we use nine different kernels, three kernels for each eye corner, and one kernel for each corner of the eyebrows. The standard deviation in the measured location is typically about one pixel for the eye corners and filtering over time reduces the error to less than 0.5 pixels. The mouth corners are measured with a similar precision, except when the mouth is very wide open so that the corner is essentially a straight vertical line. Then the vertical position may be less accurate; however, the horizontal position is still accurate. Feature points we measure in this way are mouth, nostrils, eyes, and eyebrows. Knowing precisely the positions of the eye corners and the nostrils allows an accurate determination of the head pose. The locations of the eyebrows help identifying emotions, while precise locations of the mouth corners are needed for selecting the mouth shapes for articulating speech. Sometimes the interior of the mouth is also analyzed in this way, as well as the outer edges of the lips in the center of the mouth, in order to get a better measure of lip protrusion and for estimating the stress put on the lips.

E. Pose Estimation

We apply a pose estimation technique reported in [32] (and in [38] for the coplanar version), using six feature points in the face. The four eye corners and the two nostrils have been localized with precision from a sample image (using the techniques described in Section IV-B and Section IV-C) and their relative 3-D position is known in the head model [Fig. 5(b)]. The pose estimation process starts with the assumption that all points of the 3-D model lie in a plane parallel to the image plane

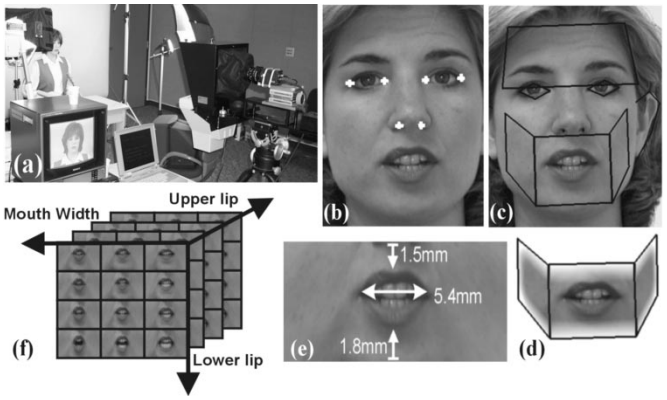

Fig. 5. Process of acquiring the head model. (a) A subject is recorded while speaking. (b) Facial features are located. (c) Head pose is calculated. (d) Bitmaps are extracted. (e) Bitmaps are normalized and features are measured. (f) Database is created.

TABLE II
VALUES SHOWN IN THE TABLE ARE THE *MAXIMUM* ERRORS IN THE CALCULATED POSE (x, y, z ANGLES IN DEGREES AND DISTANCE TO CAMERA IN mm) FOR PERTRUBATIONS OF THE MEASURED FEATURE POINTS BY: 0.5, 1, 1.5 AND 2 PIXELS. THE LAST COLUMN SHOWS AN *AVERAGE* ERROR FOR A PERTURBATION OF 1 PIXEL. THE SUBJECT WAS AT A DISTANCE OF 1 m FROM THE CAMERA. THE CAMERA FOCAL LEGNTH WAS 15 mm AND ITS RESOLUTION 560 × 480 PIXELS

Error of feature points (in pixels)	0.5	1.0	1.5	2.0	1.0 (AVG)
X-angle [deg]	1.6	3.3	5.0	6.9	0.7
Y-angle [deg]	1.4	2.8	4.3	5.8	0.67
Z-angle [deg]	0.6	1.2	1.9	2.6	0.27
Z-pos [mm]	8.9	18.3	27.4	36.6	5.6

(corresponds to an orthographic projection of the model into the image plane plus a scaling). Then, by iteration, the algorithm adjusts the 3-D model points until their projections into the image plane coincide with the corresponding localized image points. The pose of the 3-D model is obtained by solving iteratively the following linear system of equations:

$$\begin{cases} \boldsymbol{M_i} \bullet \dfrac{f}{Z_0}\boldsymbol{i} = x_i(1+\varepsilon_i) - x_0 \\ \boldsymbol{M_i} \bullet \dfrac{f}{Z_0}\boldsymbol{j} = y_i(1+\varepsilon_i) - y_0. \end{cases}$$

M_i is the position of object point i, $\boldsymbol{i}$ and $\boldsymbol{j}$ are the two first base vectors of the camera coordinate system in object coordinates, f is the focal length, and $\boldsymbol{Z}_0$ is the distance of the object origin from the camera. $\boldsymbol{i}$, $\boldsymbol{j}$, and $\boldsymbol{Z}_0$ are the unknown quantities to be determined. x_i, y_i is the scaled orthographic projection of the model point i, x_0, y_0 is the origin of the model in the same plane, ε_i is a correction term due to the depth of the model point. ε_i is the parameter that is adjusted in each iteration until the algorithm converges.

This algorithm is stable, also with measurement errors, and it converges in just a few iterations. If the recognition module has failed to identify eyes or nostrils on a given frame, we simply ignore that frame during the model creation process. The recognition module marks the inner and outer corners of both eyes, as well as the center of the nostrils. The location of the nostrils is very reliable and robust. We are able to derive their position with subpixel accuracy by applying low-pass filtering on their trajectories. The location of the eye corners is less reliable because their positions change slightly during closures. We therefore ignore frames on which the eyes are closed. The errors in the filtered positions of these feature points are typically less than one pixel. A study of the errors in the pose resulting from errors of the recognition is shown on Table II. All possible combinations of recognition errors are calculated for a given perturbation (with six points and nine possible errors, all $9^6 = 531\,441$ poses have been computed).

F. Sample Extraction

Once the head pose on an image is known, the set of polygons (quadrilaterals) marking the 3-D shape of a facial part is projected onto the image plane [Fig. 5(c)]. The projected quadrilaterals mark the boundary of each facial part on the image plane [Fig. 5(d)]. The pixels within these areas are "un-warped" using bilinear interpolation and are combined into a rectangular bitmap [Fig. 5(e)] that is now considered normalized. The resolution of these bitmaps should be adapted to the target output. In order to preserve a maximum of details, we keep the number of pixels in the sample bitmaps roughly similar to that of the corresponding area in the original image (for example: 208 × 168 pixels for the mouth area). These bitmaps can be compressed using standard compression techniques such as JPEG and MPEG2 before being saved into a database. In this way, one minute of speech for the "mouth" facial part results in an MPEG2 file of size 12 MB (208 × 168 pix at 60 frames/s) with very little loss in quality.

G. Sample Parameterization

Once all samples of a face part are extracted and stored as bitmaps, several features are computed and attached to the samples to enable fast access and selection from the database during synthesis. We use the following features.

1) Geometric Features: These are features derived from the type, relative position and size of facial parts present in the sample image. They are obtained from the facial locator module (see sections Sections IV-C and IV-D) and are transformed to correspond to a normalized, frontal view of the head. (see Section IV-F). In Fig. 5(e), for example, we describe the mouth with three parameters: the width (the distance between the two corner points), the y-position of the upper lip (the y-maximum of the outer lip contour), and the y-position of the lower lip (the y-minimum of the outer lip contour). Samples of other facial parts are parameterized in a similar way.

2) Pixel-Based Features: They are calculated from the sample pixels using principal component analysis (PCA). While geometric features described above are useful to discriminate among a large number of samples of a given facial part, they cannot capture fine details of the appearance, making it necessary to proceed with a lower-level analysis. To compute PCA, the luminance values of the normalized bitmaps are first sub-sampled in both directions into a vector of N pixels and the $N \times N$ covariance matrix is obtained by processing

all bitmaps. Then, from the singular value decomposition of the covariance matrix, the eigenvectors with the highest eigenvalues are retained. Finally, each sample is projected onto each of the principal eigenvectors to obtain a feature vector. For a set of 208 $\times$ 168 pixels "mouth" facial parts, we subsample the bitmap by a factor of six and we keep 30 components that account for over 95% of the variance in the input images. Misalignment of samples (offsets, rotations, etc.) would prevent PCA to account for most of the variance with a small number of principal components and, hence, would not be useful here. However, since the samples have been normalized (the head pose is factored out), PCA components are very good features for discriminating fine details in the shapes of facial parts.

3) Head Pose: These are the angles and the position of the head in the given image. This information is saved for each image that is used as base face. It is used at synthesis time to project facial parts samples onto the base face with the correct orientation.

4) Original Sequence and Frame Number: These are the frame number and sequence number in the original recorded video data. This information is used when selecting samples for an animation to preserve whenever possible the inherent smoothness of real, recorded facial movements.

5) Phonetic Information: This information is obtained using automatic speech recognition techniques and consists of a DARPA-bet phone symbol attached to each frame.

H. Database of Samples

First, the space defined by the geometric features of a facial part is quantized at regular intervals. This creates an $\boldsymbol{n}$-dimensional lattice, where n is the number of parameters [Fig. 5(f)], and each lattice point represents a particular appearance of the facial part. Going through all samples, each one is inserted in the closest lattice entry (smallest Euclidian distance in the feature space). Next, the samples of each lattice entry are clustered using pixel-based features. We use a simple two-pass leader cluster algorithm and set the cluster distance threshold to a small value. This has the effect of clustering samples that look quasi-identical and, thus, reduces drastically the size of the database without reducing the diversity of appearances within one lattice point. The number of samples kept in each lattice point varies, with some being empty. Having multiple samples per lattice point is necessary because the geometric features often fail to completely characterize the appearance of a facial part. For example, the visibility of the teeth and the tongue in the mouth cavity are rather difficult to analyze and, if taken as geometric features, would substantially increase the dimensionality of the lattice. This would multiply the number of samples in the database. Instead, we rely on pixel-based features to capture these subtle differences.

There is a tradeoff between the size of the database and the quality of the animations that it can generate. Being more aggressive in clustering samples within lattice points results in a smaller database of samples; however, this will tend to produce animations that appear more jerky (see Section V). For example, in one of our experiments, we recorded approximately 3 min of speech, resulting in about 40 MB of mouth samples (see Section IV-F). After clustering, the size of the database was approximately reduced by half without noticeable loss in the quality of the animations.

TABLE III
FROM THE INPUT TEXT "I BET THAT," PHONEMES ARE OBTAINED FROM TTS AND THE COARTICULATION OF FACIAL PARAMETERS IS COMPUTED. FOR EACH FRAME OF THE ANIMATION, THE TABLE SHOWS THE VALUES OF EACH PARAMETER

frame number	phoneme	mouth width	lower lip pos.	upper lip pos.
44	ah	70	51	77
45	ah	71	55	79
46	ah	80	50	77
47	iy	88	44	74
48	iy	87	36	78
49	b	79	24	88
50	b	74	19	93
51	b	72	34	89
52	eh	69	60	82
53	eh	67	78	76
54	eh	70	71	62
55	eh	77	53	42
56	t	83	44	31
57	dh	85	48	30
58	ae	86	51	35
59	ae	91	54	53
60	ae	95	58	74
61	ae	95	53	67
62	ae	93	41	48
63	t	89	32	32
64	t	84	29	27

V. SYNTHESIS

A. Audio Input

Talking-head animations are generally driven by audio input, whether in the form of recorded audio from a subject speaking [20], or from synthesized audio [33]. Using a text-to-speech synthesizer (TTS) to provide the audio makes for a fully automated system that can synthesize talking-head animations from ASCII text input. The drawback of using a TTS is that most of them produce speech with a distinctly robot-like sound. Subjective tests show [4] that users dislike the combination of photo-realistic video and synthetic-sounding audio (actually preferring synthetic-looking video combined with synthetic-sounding audio). Since we are striving for natural appearance of our face, we were searching for a TTS that sounds natural. Only very recent progress in speech synthesizer technology has produced speech that can be considered naturally sounding [6]. Tests with one of these TTS systems confirmed that its naturalness is sufficient to accompany a recorded face [4]. Starting from ASCII text input, the TTS produces a sound file along with its phonetic transcription. This transcript includes precise timing information for each phoneme (Table III). To control other facial parts during the production of speech, special marks are placed within the text input that are recognized by the TTS, which then produces timing information for these marks.

B. Animation of the Mouth

To animate the mouth from a phonetic transcript, the naïve approach of mapping each phoneme to a particular mouth shape leads to very poor articulation. This is because of the coarticulation effect. When we articulate a phoneme, the lips, jaw,

and tongue either get ready ahead of time to articulate the next phoneme(s) (as in "stew," the lip protrusion starts appearing before the "u" sound comes out), or linger on from the previous phoneme(s) (as in "yult," the lip protrusion stays after the "u" sound stops). Another commonly used model for coarticulation is the Cohen–Massaro model [22], [37], which calculates parameters of the mouth (such as upper lip position, lip width, jaw rotation, tongue, etc.) of a given frame by integrating weighted contributions of participating phonemes. This model was originally developed to drive a 3-D polygonal model and is not well adapted to our sample-based approach. Instead, we also use a sample-based approach for coarticulation. Using the phonetic features, the best matching frames to the current phonetic context are found and their associated geometric features are averaged. For each frame of the animation, the resulting feature vectors create a trajectory in the space of features. Fig. 6 shows a trajectory in the space of three mouth parameters shown on Table III for the utterance: "I bet that." A 3-D space is shown here for simplicity; however, we typically include other mouth parameters such as jaw rotation, tongue, and teeth visibility.

To create a video animation, the trajectory is sampled at the video rate (typically every 33.33 ms for a 30 frames/s animation). Then, for each sample point, the closest lattice entry of geometric features is chosen, providing a set of candidate mouth bitmaps. A graph is then constructed for the animation (Fig. 7) containing a list of candidate mouth bitmaps for each video frame. Transition costs are calculated between candidates of two consecutive frames. This cost is either the Euclidian distance in PCA space between the two candidate images or zero for frames that are following each other in the original recordings (actually within k frames of each other with k typically equal to 3)

$$cst(bmp1, bmp2) = \begin{cases} 0, & \text{IF } \begin{cases} 0 \le fr(bmp1) - fr(bmp2) < k, \\ \text{AND} \\ seq(bmp1) = seq(bmp2) \end{cases} \\ g(bmp1, bmp2), & \text{IF } \begin{cases} fr(bmp1) - fr(bmp2) \ge k, \text{ OR} \\ fr(bmp1) - fr(bmp2) < 0 \text{ OR} \\ seq(bmp1) \ne seq(bmp2) \end{cases} \end{cases}$$

$$\text{with } \begin{cases} g(bmp1, bmp2) = \sqrt{\sum_{i=1}^{k} (PCA_{bmp1,i} - PCA_{bmp2,i})^2} \\ fr(bmp1) = \left| \begin{array}{l} \text{frame number of} \\ \text{bitmap in sequence} \end{array} \right. \\ seq(bmp1) = \left| \begin{array}{l} \text{sequence number} \\ \text{to which the} \\ \text{bitmap belongs} \end{array} \right. \end{cases}$$

Once the transition costs between every node have been established, the shortest path through the graph is computed using a Viterbi search. The resulting least expensive path corresponds to the smoothest possible animation, the one where visual differences between consecutive frames are minimized. To keep computation costs low, we limit the number of candidates at each lattice point to 50.

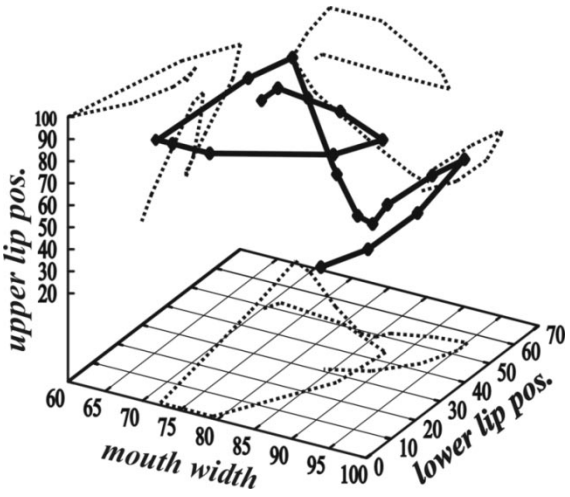

Fig. 6. From a phonetic transcript, parameters of the mouth are calculated and a trajectory is obtained in the space of parameters that is sampled to obtain an animation script.

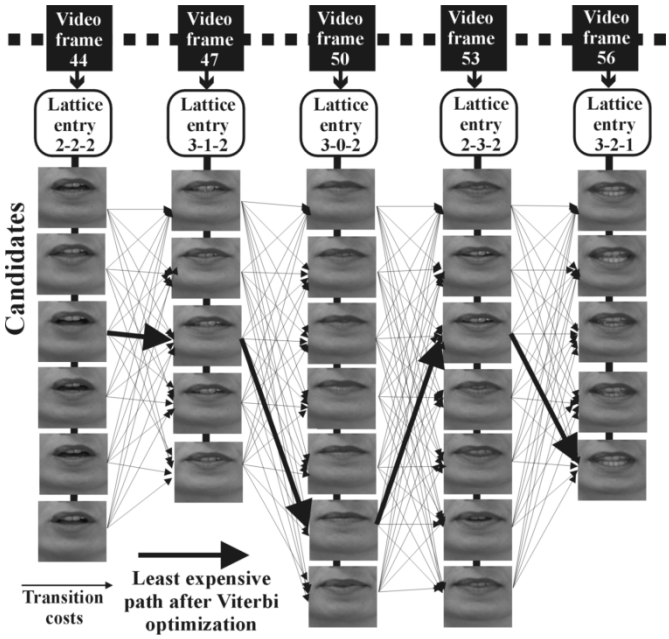

Fig. 7. Using the animation script: for each frame a lattice entry in the database is chosen, providing a list of candidate bitmaps. Transition costs are computed in the PCA domain between each candidate of each frame. A low cost indicates that the two bitmaps are visually close. A Viterbi search is then performed to find the path that incurs the lowest cost and hence generates the smoothest animation.

C. Transitions

When transition costs between consecutive frames are high, indicating a large visual difference, we attempt to smooth the transition by gradually merging one frame into the other using alpha-blending:

$$pix_{i,j} = \alpha \cdot pixa_{i,j} + (1 - \alpha) \cdot pixb_{i,j}$$
$$\alpha = \frac{t - t_0}{t_1 - t_0} \qquad t \in [t_0, t_1].$$

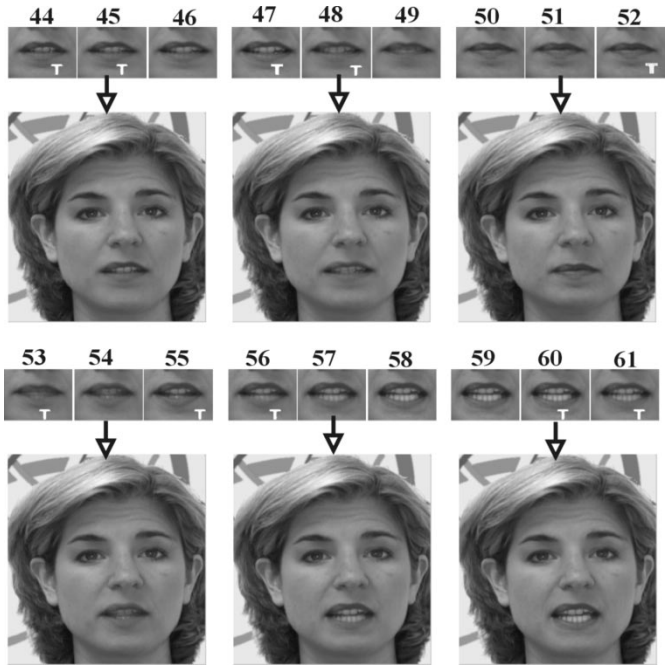

Fig. 8. Mouth bitmaps selected for the animation. Each bitmap is then inserted into the whole head for the final animation.

During the transition interval from t_0 to t_1, the resulting pixel *pix* is a blend of the corresponding pixels from sample **a** $(pixa)$ and sample **b** $(pixb)$. The number of samples that are used to create a transition varies depending on the sampling rate of the trajectory, the duration of the samples, and the transition cost. To enhance the quality of transition frames and avoid the occasional see-through effect of alpha-blending, we have experimented with a morphing technique described in [10]. However, we found that for the mouth, alpha-blending is often preferable. While the quality of the transition images is excellent, morphing is computationally more expensive. Furthermore, morphing tends to introduce coarticulation artifacts. Morphed images are not necessarily real occurrences of mouth shapes and hence might distort the perceived lip-synch. For other facial parts, however, morphing provides better results than blending. Fig. 8 shows the sequence of mouth shapes selected from the database, plus the transition shapes, marked with a white T.

D. Other Facial Parts

We handle the animation of other facial parts using a model similar to the one developed for the MPEG4 facial animation subsystem [12]. Special markers are put in the text to control amplitude, length, onset, and offset of facial animations. This is an easy way to provide synchronization of conversational cues, such as eye and eyebrow movements, eye blinks, or head movements that accompany the spoken text.

As for the base face, we use a set of recorded sequences of speech, where the subject was asked to speak while keeping movements of the jaw to a minimum. In this way, we avoid artifacts that sometimes appear when overlaying a closed mouth (small footprint) over a base face with an open mouth (large footprint). These sequences otherwise exhibit the usual characteristics of speech, including slight head movements, eye blinks, and occasional eyebrows movements. Using these sequences as a substrate for entirely different speech might seem objectionable, since the synchronization of head movements and other facial parts movements will be lost. However, we found that for short speech animations, people mostly pay attention to the lip-synch and never complained about unsynchronized head movements.

E. Rendering

To synthesize a new face with a certain mouth shape and emotional expressions, the proper sample bitmaps are chosen for each of the facial parts. For example, at a given time of an animation, the talking-head is supposed to utter an “u” with emphasis, with the head slightly lowered. The coarticulation module selects the proper sample bitmap from the database for the mouth (here with the protruted lips typical of the “u” sound), the eyes (here wide open for emphasis), and the forehead (here, risen eyebrows for emphasis again). Then a base face is selected from the database (here slightly lowered). The bitmap of the base face is first copied into the frame buffer, then the bitmaps of face parts are projected onto the base face using the 3-D model and the pose of the base head. Because we limit ourselves to frontal views with small movements, no overlap or occlusion of facial parts occur. To avoid any boundary artifacts from overlaying bitmaps, we use gradual blending or “feathering” masks [Fig. 1(c)]. These masks are created by ramping up a blending value from the edges toward the center.

F. Synthesis Performance

The overall system is currently able to produce video files (AVI compressed in MJPEG) from ASCII text input without any manual intervention. The text is first sent to the TTS module, which produces an audio file and a phoneme file. The Synthesis module then produces the video file. At the moment, with no special care given to optimizing the system for speed, we are able to produce video at about five frames/s with a latency of 1/100 s/frame (hence, a 10–s animation at 30 frames/s will take about 63 s : 3 s initial latency + 60 s for producing the video). The latency comes from the TTS module and the Viterbi search, and half of the rendering time is spent compressing the video. Since the actual rendering (projecting and alpha-blending the sample bitmaps) can be done efficiently on most computers using OpenGL texture-mapping functions, we expect to have a real-time system in the near future.

VI. Results and Discussion

We have produced short videos from several different head models. Fig. 8 shows a few frames extracted from such a video. It is, of course, impossible to judge the quality of an animation from still pictures; this only shows that statically these frames look natural with no noticeable artifacts. To obtain feedback on the quality of these sequences, we have made informal tests with dozens of people. All tests were done with short clips, without integrating them into an application or trying to make them particularly entertaining (such as telling a joke). In such a setting, the viewers concentrate fully on the talking head and notice any artifacts. While reactions are mostly positive, some viewers criticize the lip synchronization

and the articulation—often over-articulation. Occasionally modeling or blending artifacts at the teeth and the jaw are visible. The reader is encouraged to go to our web site at http://www.research.att.com/∼eric/synthesis.html to view some examples of talking head animation generated by our systems.

A formal test was done to determine whether a talking head could improve intelligibility of spoken text in a noisy environment [4]. Two head models were tested, one 3-D model, with and without texture maps of a real person, and this sample-based head. All head models improved intelligibility significantly and by about the same amount (error rate dropped from 20% for audio-only to only 4% for the talking-head). These tests were done with an older version of the sample-based talking head and all heads used an older TTS [13], which has more of a robot- like voice. Subjective tests indicated that users dislike the clash between a natural-looking face and an unnatural voice (in contrast, a synthetic-looking head with a natural voice seems to be perfectly acceptable). Therefore, in some of the tests, the bare 3-D model scored higher in "being liked" than either the sample-based head or the 3-D head with a person's texture map. The new TTS [6] sounds very natural and fits well the appearance of the sample-based head. In recent tests, there have never been any complaints about a mismatch between voice and face.

A long-term goal is to produce animation snippets that cannot be distinguished from a real person, or at least to make the animations look so good that a viewer accepts them as a replacement for video clips of a person. In order to make a talking head a valuable addition to an application, it is not only its appearance that must be of very high quality. In fact, to keep viewers pleased, the talking head must have a wide repertoire of behaviors blending discreetly into the flow of action of the surrounding application.

A. Model

The simple 3-D model we currently use covers a limited range of views. This is because it approximates facial parts with only a few planes, resulting in increasingly visible artifacts when the head rotates away from the frontal position. There is no simple solution to this problem and, in fact, our system takes advantage of the fact most speech postures are frontal with small rotations around the frontal position. To circumvent this limitation, a more tightly fitting 3-D model is needed and textures need to be sampled from multiple directions, resulting in a complex capture process and a large database.

There is an incentive, on the other hand, to keep the modeling part simple enough so that it can go outside the lab into "modeling shops." These shops are places where people can get their model made for a fee. Cyberware scanner shops are becoming increasingly common, showing people's growing interest in getting their own digital representation. Data-driven techniques are inherently cumbersome, because of the large amount of data that has to be recorded from a subject to create his or her model. However, we are working toward keeping the overall cost of the modeling to a minimum through a simple and nonintrusive capture process (single camera, no audio, and no facial markers or head restraints).

B. Emotions

Even though our talking-head model exhibits striking photo-realism, it still lacks prosody, emotions, and synchronized head movements. Though this is acceptable for short animations (15 s or less), longer animations quickly become boring to watch. Currently, emotional expressions are generated mostly through animations of the upper part of the face (eyebrows raising and frowning, eyes widening, etc.). We plan to record additional sets of the mouth where the subject is smiling while talking. By switching between the happy and the regular set of mouths, we hope to be able to convey some degree of emotional content in the speech, making long animations more enjoyable. We also plan to produce synchronized head movements by controlling morphing within a lattice of base heads in various common positions.

VII. Conclusions

We have presented a novel way to create head models that can be used to generate photo-realistic talking-head animations. Using image samples captured while a subject was speaking preserves its original appearance. Using photographs as parts of computer graphics is an old tradition. Yet as long as photographs had to be segmented manually, this approach was very costly. Recent advances in accuracy and robustness of face recognition systems make the approach described here feasible. The pose of the head and the precise location of facial parts on tens of thousands of video frames are computed, resulting in a rich yet compact database of samples. A simple 3-D model of the head and facial parts enables perspective projection of the samples onto a base head in a given pose, allowing head movements. The results are lively animations with a pleasing appearance that resemble closely a real person. This system looks promising for generating talking heads that can enliven computer-user interfaces as well as future encounters among avatars in cyberspace. As envisioned by science fiction author Neil Stephenson in his description of the "Metaverse" [39], people will meet and interact through avatars and these avatars, through their quality, beauty, and realism (or lack thereof) will be representative of the social status, wealth, and technological savvy of their real counterparts. Models that are highly realistic, of course, come at a higher cost, namely model creation costs, and costs related to the size (bytes) of the model, costs related to the resources needed to animate the model. But these costs will be justified by the true-to-real-life experience these models can provide.

Acknowledgment

The authors would like to thank J. Ostermann for setting up subjective testings and his participation in data collection, T. Ezzat for sharing his knowledge in morphing algorithms and image PCA, G. Potamianos for many useful discussions on visual unit selection, and A. Basso for his precious help with MPEG2 video compression. The text-to-speech synthesizer used here was developed by J. Schröter, M. Beutnagel, A. Conkie, Y. Stylianou, and A. Syrdal, who provided access to timing and prosodic information and made the system available at an early stage of the development.

REFERENCES

[1] T. F. Cootes, K. Walker, and C. J. Taylor, "View-based active appearance models," in *Proc Int. Conf. Automatic Face and Gesture Recognition*, 2000, pp. 227–232.

[2] *Proceedings of the 4th International Conference on Automatic Face and Gesture Recognition*. Los Alamitos, CA: IEEE Comput. Soc. Press, 2000.

[3] H. P. Graf, E. Cosatto, and T. Ezzat, "Face analysis for the synthesis of photo-realistic talking-heads," in *Proc Int. Conf. Automatic Face and Gesture Recognition*, 2000, pp. 189–194.

[4] I. Pandzic, J. Ostermann, and D. Millen, "User evaluation: Synthetic talking faces for interactive services," *Vis. Comput.*, vol. 15, no. 7/8, pp. 330–340, Nov. 04, 1999.

[5] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3D faces," in *Proc. SIGGRAPH'99*, July 1999, pp. 353–360.

[6] 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.

[7] J. J. Lien, T. Kanade, J. F. Cohn, and C. C. Li, "Detection, tracking and classification of action units in facial expression," *J. Robot. Auton. Syst.*, 1999.

[8] B. Guenter, C. Grimm, D. Wood, H. Malvar, and F. Pighin, "Making faces," in *Proc. SIGGRAPH'98*, July 1998, pp. 55–66.

[9] F. Pighin, J. Hecker, D. Lichinski, R. Szeliski, and D. H. Salesin, "Synthesizing realistic facial expressions from photographs," in *Proc. SIGGRAPH'98*, July 1998, pp. 75–84.

[10] T. Ezzat and T. Poggio, "Miketalk: A talking facial display based on morphing visemes," in *Proc. Computer Animation*, June 1998, pp. 96–102.

[11] E. Cosatto and H. P. Graf, "Sample-based synthesis of photo-realistic talking-heads," in *Proc. Computer Animation*, June 1998, pp. 103–110.

[12] J. Ostermann, "Animation of synthetic faces in MPEG-4," in *Proc. Computer Animation*, June 1998, pp. 49–55.

[13] J. Olive, J. Van Santen, B. Boebius, and C. Shih, "Synthesis," in *Multinlingual Text-to-Speech Synthesis*, R. Sproat, Ed. Norwell, MA: Kluwer, 1998, ch. 7, pp. 191–228.

[14] J. Lengyel, "The convergence of graphics and vision," *IEEE Computer*, vol. 31, no. 7, pp. 46–53, July 1998.

[15] G. Potamianos, H. P. Graf, and E. Cosatto, "An image transform approach for HMM based automatic lip reading," *Proc. IEEE Int. Conf. on Image Processing*, vol. III, pp. 173–177, 1998.

[16] F. I. Parke and K. Waters, *Computer Facial Animation*. Wellesley, MA: A. K. Peters, 1997.

[17] M. Escher and N. Magnenat-Thalmann, "Automatic 3D cloning and real-time animation of a human face," in *Proc. Computer Animation*, 1997, pp. 58–66.

[18] A. Lanitis, C. Taylor, and T. F. Cootes, "Automatic interpretation and coding of face images using flexible models," *PAMI*, vol. 19, no. 7, 1997.

[19] I. A. Essa and A. P. Pentland, "Coding, analysis, interpretation and recognition of facial expressions," *PAMI*, vol. 19, no. 7, 1997.

[20] C. Bregler, M. Covell, and M. Slaney, "Video rewrite: Driving visual speech with audio," in *Proc. SIGGRAPH'97*, July 1997, pp. 353–360.

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

[22] D. W. Massaro, *Perceiving Talking Faces*. Cambridge, MA: MIT Press, 1997.

[23] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*. Cambridge, U.K.: Cambridge Press, 1997.

[24] D. DeCarlo and D. Metaxas, "Optical flow constraints, on deformable models with applications to human face shape and motion estimation," in *Proc. CVPR*, 1996, pp. 231–238.

[25] D. Beymer and T. Poggio, "Image representation for visual learning," *Science*, vol. 272, no. 28, pp. 1905–1909, June 1996.

[26] S. M. Seitz and C. R. Dyer, "View morphing," in *Proc. SIGGRAPH'96*, July 1996, pp. 21–30.

[27] M. Bichsel, "Automatic interpolation and recognition of faces by morphing," in *Proceedings of the 2nd International Conference on Automatic Face and Gesture Recognition*. Los Alamitos, CA: IEEE Comput. Soc. Press, 1996, pp. 128–135.

[28] 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.

[29] M. J. Black and Y. Yacoob, "Recognizing facial expressions under rigid and nonrigid facial motions," in *Proc Int. Conf. on Automatic Face and Gesture Recognition*, 1995, pp. 12–17.

[30] A. Saulnier, M. L. Viaud, and D. Geldreich, "Real-time facial analysis and synthesis chain," in *Proc Int. Conf. on Automatic Face and Gesture Recognition*, 1995, pp. 86–91.

[31] Y. Lee, D. Terzopoulos, and K. Waters, "Realistic modeling for facial animation," in *Proc. SIGGRAPH'95*, July 1995, pp. 55–62.

[32] D. Dementhon and L. Davis, "Model-based object pose in 25 lines of code," *Int. J. Comput. Vis.*, vol. 14, pp. 123–141, 1995.

[33] K. Waters and T. Levergood, "DECface: A system for synthetic face applications," *Multimedia Tools Applicat.*, vol. 1, no. 4, pp. 349–366, 1995.

[34] A. Pentland, B. Moghaddam, and T. Starner, "View-based and modular eigenspaces for face recognition," *CVPR*, 1994.

[35] C. Pelachaud, N. I. Badler, and M. L. Viaud, "Final report to NSF of the standards for facial animation workshop," Univ. Pennsylvania, Philadelphia, 1994.

[36] P. Fua and Y. G. Leclerc, "Using 3-dimensional meshes to combine image-based and geometry-based constraints," in *Eur. Conf. Computer Vision*, May 1994, pp. 281–291.

[37] M. M. Cohen and D. W. Massaro, "Modeling coarticulation in synthetic visual speech," in *Models and Techniques in Computer Animation*, M. Magnenat-Thalmann and D. Thalmann, Eds. Tokyo, Japan: Springer Verlag, 1993.

[38] D. Oberkampf, D. Dementhon, and L. Davis, "Iterative pose estimation using coplanar feature points," Univ. Maryland, College Park, Intern. Rep., CVL, CAR-TR-677, July 1993.

[39] N. Stephenson, *Snow Crash*. New York: Spectra, June 1992.

[40] G. Wolberg, *Digital Image Warping*. Los Alamitos, CA: IEEE Comput. Soc. Press, 1990.

[41] G. Faigin, *The Artist's Complete Guide to Facial Expression*. New-York, NY: Watson-Guptill, 1990.

[42] P. Ekman and W. Friesen, *Facial Action Coding System: A Technique for the Measurement of Facial Movement*. Palo Alto, CA: Consulting Psychologists, 1978.

[43] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimal decoding algorithm," *IEEE Trans. Inform. Theory*, vol. IT-13, 1967.

Eric Cosatto received the M.S. degree in computer science from the Swiss Federal Institute of Technology, Lausanne, Switzerland, in 1991.

He is a Senior Research Scientist at AT&T Labs-Research, Red Bank, NJ. His main areas of interest are computer graphics, computer user interfaces, image understanding, and image processing.

Hans Peter Graf (M'86–SM'89–F'96) received the Diploma in physics in 1976 and the Ph.D. degree in physics in 1981, both from the Swiss Federal Institute of Technology, Zürich, Switzerland.

He is a Technology Consultant at AT&T Labs-Research, Red Bank, NJ, leading research projects on machine vision and animated computer characters. He published over 100 scientific papers and holds 11 patents, with several more pending.

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