In recently developed interactive virtual environments, video and audio are the most fundamental elements and important as input methods. Video input is used not only for recording and replay, but also for the analysis and recognition of objects and motion. Audio input is an equally intuitive medium for communication with the virtual humans. These media enable an operator to control the environment intuitively without using a special device. For such a situation, we have developed a complete system that covers the aspects from the cloning of a face to its video and audio driven animation. This makes it possible to make a virtual environment rapidly and easily and act in it through the video and the audio input.

Technologies to create a cloned face by deforming a generic three-dimensional (3-D) head model, to create the texture from two or three photos, and its animation have been developed by some researchers [4], [5]. Yet these methods are to be automated and to make a cloned head is still a tedious job. Thus, we had to work out a method to recognize features on an individual’s face to specify the location of each feature. The researchers are working in facial analysis or facial feature extraction to help feature extraction using template matching [6], [7], feature classification [8], knowledge-based system, neural network system, maximum likelihood [9], [10], etc., and each method has its strong and weak points.

Tracking of facial expressions is also researched using many methods. For instance, DeCarlo et al. [12] have applied optical flow and a generic face model based algorithm. This method is robust but is not suitable for a real-time system. Cosatto et al. [13] use a sample-based method that needs to make a sample for each person. Kouadi et al. [14] use a database and some face markers to process it effectively in real time. However, the use of markers is not always practical, and it is more attractive...
Our methodology to construct a cloned face from two orthogonal pictures is based on the idea of using the uniqueness of a human.

to recognize features without them. Pandzic et al. [15] use an algorithm based on edge extraction in real time. Our animation system is based on the MPEG-4 standard and is aimed to animate in real time by tracking the actor's face.

To realize an expressive clone in a virtual environment, the speech animation is very important. For this, it is necessary to extract the phonemes directly from the speech signal and apply them to the synthetic 3-D face with lip synchronization for the complete facial animation of the clone or the avatar. Various efforts have been made for extracting different parameters from the speech signal and mapping them to the mouth/ lip parameters. McAllister et al. [16] used mouth shape descriptors called moments computed from the fast Fourier transform coefficients of the speech signal. They concluded that these moments are correlated with the position of the mouth. The jaw position and the vertical and the horizontal distance between the lips extracted from the associated video were used for this purpose. Curinga et al. [17] used similar mouth shape parameters extracted from the recorded video sequences. They used the cepstral coefficients obtained from speech analysis. In addition, time delay neural networks were used to overcome the problem of co-articulation. Yamamoto et al. [18] used 3-D position sensors to acquire the visual data for training, and Brand [19] developed a voice puppetry system for generating facial animation from expressive information in an audio track. This involved integration of various sophisticated techniques in image processing, computer vision, and speech analysis. These approaches typically involve numerically complicated algorithms and extensive audiovisual data collection and training phase critical for system performance. The results obtained are very realistic; however, such systems may not be suitable for real-time applications.

Figure 1 shows our overall system with different tasks and interactions required to generate a real-time avatar or a mimicking clone in a shared or networked virtual environment [1]. Two photos of the actor’s face are taken: one from the front and the other from the side to make the clone before the animation process. The images are processed by automatic face feature extraction system and the virtual face is created. The video input and the speech of the user drive the facial animation of the clone. In this process, MPEG-4 facial animation parameters (FAPs) are extracted in real time from the video input of the face. These FAPs can then be used either to animate the cloned face or to compute the high-level emotions transmitted to the autonomous actor. The phonemes can be extracted from the speech. The phonemes are then blended with the FAPs from the video to enhance the animation of the clone.

The FAPs require low bandwidth to be transmitted [2], [3] so that it is possible for real-time animation as it will be described later.

In this article, we describe the components of the system used for real-time facial communication using a cloned head. We begin with describing the automatic face cloning using two orthogonal photographs of a person. The steps in this process are the face model matching and texture generation. After a brief introduction to the MPEG-4 parameters that we are using, we proceed with the explanation of the facial feature tracking using a video camera. The technique requires an initialization step and is further divided into mouth and eye tracking. These steps are explained in detail. We then explain the speech processing techniques used for real-time phoneme extraction and subsequent speech animation module. We conclude with the results and comments on the integration of the modules towards a complete system.

Automatic Face Cloning

The major difficulty in facial feature extraction is the inherent variability of the image formation process in terms of image quality, environmental condition, and the variety of human races, hair styles, etc. On the other hand, human vision system, compared with an artificial vision, can easily build a human face from such information as motion, color, orientation, and edge. These features establish an attentive representation on which the visual attention is based. The evaluation of this representation will provide different salient image regions.

Our methodology to construct a cloned face from two orthogonal pictures is based on the idea of using the uniqueness of a human. The two pictures we use do not convey motion information but edge information, and sometimes color information. Human vision system may construct a face from this kind of image by the combination of the segment division of the organs on the face and the detail region recognition after the segmentation:
The segmentation of the face is based on knowledge of the human face. Facts like “there are two eyes around the center of the face,” or “the mouth is below the nose” help the segment construction (knowledge-based recognition).

The detail recognition is based on the combined information of edge appearance, color difference, and so on (i.e., feature-based method).

To mimic this human visual system, global matching of a generic face is used to find an approximate face construction at the first stage. After this process, detail feature extraction of each feature is done by an effective method for each organ.

**Face Model Matching**

A generic 3-D model of a human face is used to divide a face into segments. In this process, we separate the model into several parts to match features on the face. At the beginning of the model matching, the global position matching of all the parts is done, thus we call this method global matching. This global matching helps in avoiding feature position matches to noise like a wrinkle or a moustache. Figure 2 shows a generic face model used in this system. The first image (a) is the whole face model in which every feature region mask is connected. At this step, the face picture and the generic face model does not fit at all. The 3-D model is not made from the mean face structure. The second image (b) shows the model separated into several feature regions like the eyes, the nose, the eyebrows, the mouth, and the jaw region, and the global positions are moved to the precalculated mean position. In the third image (c), the final result that these regions fit to the proper position by the matching algorithm is shown. To fit these regions, we first used Canny edge detector to extract the edges of the face and choose the horizontal edges. This is because almost every feature on the face, like the eyes and the mouth, is based on horizontal lines. Then, the position of each feature is guessed by combining the weight made from the standard distribution data and the edge strength. Image (d) is the result after the detail fitting is done. In this process, we used different algorithms for each region. For example, at the forehead line and the jaw line, image preprocessing is done for enhancing the edge information because these lines are very weak sometimes or have much noise nearby. After this process, the energy-based line fitting algorithm, similar to snake method, and symmetry information are used to fit to the correct position. The side view of the face is also processed by a similar method. The generic model (e) is deformed to fit to the side view of the face. Image (f) is the result of the side view facial feature extraction after the detail fitting. Hence, after the extraction of the front and the side facial features, the 3-D face model is created.

2. Automatic face feature detection.
automatically by deforming a generic model. Image (g) shows the deformed generic model made from the front and side facial features, and image (h) shows the final result with putting the face texture, generated from the two images, on the model [2].

**MPEG-4 in Facial Animation**

A detailed discussion on the MPEG-4 standard is beyond scope of this article, and we refer to [11]. Since we are interested in the facial animation primarily, we explain in brief the MPEG-4 parameters related to facial animation here.

**Face Definition Parameters**

The Face and Body Animation Ad Hoc Group (FBA) has defined in detail the parameters for both the definition and animation of human faces and bodies. Face definition parameters are defined in terms of feature points on the key locations of the face, e.g., tip of nose, hair boundary, etc.

**Facial Animation Parameters**

These parameters are designed to cover the naturally possible as well as exaggerated expressions. The animation parameters are defined precisely to allow an accurate implementation on any facial model. There are 68 FAPs on the face to make an expression, and each parameter indicates the normalized movement of the feature point in a defined direction.

All the FAPs involving translational movement are expressed in terms of facial animation parameter units. These units are defined in order to allow the interpretation of FAPs on any facial model in a consistent way, producing reasonable results in terms of expression and speech pronunciation. They correspond to fractions of distances between some essential facial features (e.g., eye distance, length of nose, etc.).

**Facial Feature Tracking**

To obtain a real-time tracking, many problems have to be solved. One important problem lies in the variety of the appearances of individuals, such as skin color, eye color, beard, moustache, glasses, etc. The second problem comes from the camera environment. The main purpose of tracking in real time is to get as much information at the initialization phase as possible and reduce the computation overhead in the tracking phase. Thus, after the automatic face detection, the whole related information of the actor’s face is extracted automatically. The tracking is robust for normal condition but weak at changing of lighting condition.

**Initialization**

In this application, the face features and their associated information are set during an initialization phase that will solve the main problem in facial feature differences between people. Figure 3(a) shows this initialization. In this phase, the same technique that is used to make a cloned face is used. The program automatically recognizes salient regions to decide the initial feature positions. Therefore, no manual manipulation is required and the information around the features, the edge information,
and the face color information are extracted automatically. Color pixel, neighbor relation, and edge parameters used during feature tracking are then generated automatically. Those parameters, gathered during initialization phase for every face, contain all the relevant information for tracking the face position and its corresponding facial features without any marker.

The tracking process is separated into two parts: mouth tracking and eye tracking. The edge and the gray level information around the mouth and the eyes are the main information used during tracking. Figure 3(b)-(c) display two examples of the tracked features superimposed on the face images.

**Mouth Tracking**

The mouth is one of the most difficult facial features to analyze and track. Indeed, the mouth has a very versatile shape and almost every muscle of the lower face drives its motion. Furthermore, a beard, a moustache, the tongue, or the teeth might appear sometimes and further increase the difficulty in tracking. Thus, many researchers are working on lip tracking or lip reading based on image processing. But the methods used for detailed mouth shape tracking, like the methods mentioned previously, are not always useful for real-time tracking when comparing quality and speed.

Our method is taking into account some intrinsic properties of the mouth:

▲ Upper teeth are attached to the head bone, and therefore their position remains constant.

▲ Conversely, lower teeth move down from their initial position according to the rotation of the jaw joints.

▲ Basic mouth shape, open or closed, depends on bone movement.

From these properties it follows that detection of the positions of hidden or apparent teeth from an image is the best way to make a robust tracking algorithm of the mouth shape and its associated motion.

The basic flow of the algorithm for tracking the mouth is depicted in Fig. 4. The system proceeds first with the extraction of all edges crossing a vertical line going from the nose to the jaw. In the second phase, the energy that shows the probability of a mouth shape, which is based on the edge and pixel value matching combination, is calculated. Among all possible mouth shapes, the best candidate is chosen according to a highest energy criterion from the database, which contains rules of a combination between the basic mouth shape and the edge appearance.

Figure 5 presents the simple rule contained in the database. Different shapes of the mouth make different edge level values in the vertical line from the nose to the jaw. For example, an open mouth and a closed mouth have the following features:

▲ **Closed mouth:** In this case, the center edge appears strong, the other two edges appear normally weak, and teeth are hidden inside; thus, the edge is not detected.

▲ **Opened mouth:** As shown in the figure, when teeth are present, the edges are stronger than the edge on the outside lips or between a lip and the teeth, or between the lip and the inside of the mouth. If the teeth were hidden inside the lips, upper or lower, the edge of the teeth would not be detected.

Once this edge detection process is done, the extracted edge information is compared with the data from a generic shape database and a first selection of possible corresponding mouth shapes is done.

Henceforth, few top candidates are passed to the second phase. At the second phase, a mouth shape for each candidate is extracted. Figure 6 shows how the mouth shape is calculated. The edges are searched from the center, and the edges make an approximate mouth shape. It will not make a proper shape when the candidate is not a proper mouth shape. Thus, the shape probability for each mouth candidate is calculated from the edge connection. The best shape is chosen as the current mouth position. Thus, we can extract and guess the approximate mouth
shape in real time, not only from the image but using the mouth model information from a database.

Eye Tracking

Eye tracking is considered a combination of pupil position tracking and recognition of eyelid position. An extraction of an eyebrow position will also help to recognize the eye tracking. Thus, the eye tracking system includes the following subsystem: pupil tracking, eyelid position recognition, and eyebrow tracking. These three subsystems are deeply dependent on one another. For example, if the eyelid is closed, the pupil position is hidden, and it is obviously impossible to detect its position. In our first attempt, we considered an algorithm combination of a generic model of an eye and optical flow method, but the system was failing when the eye was closed, and the stabilization of the result was difficult to obtain. One major factor that caused this problem was that the motion of the eyelid was too fast to track. Thus we developed a new method based on a knowledge database, like the mouth tracking. We improve the method by:

1) Calculating both pupil positions.
2) Calculating eyebrow positions.
3) Extracting eyelid positions with respect to their possible position.
4) Checking all data for the presence of movement. After this stage, inconsistencies are checked again and a new best position is chosen if necessary.

Figure 7 presents this flow briefly.

For tracking a pupil in real time, there are some problems to be solved. Sometimes a pupil is hidden behind an eyelid. In such a case, the object to track (i.e., a pupil) is hidden immediately. Also, the motion speed of a human’s eyeball is very fast compared with the image capturing speed, 60 frames/s. Thus, if the pupil tracking has lost, there are two possibilities: one is due to the fast motion, and the other is due to the eyelid action.

At the first stage of pupil tracking, a kind of energy is used. This value is calculated from the maximum matching probability, by edge value, color similarity calculated from the image self-correlation, and position movement. But here, a problem occurs if there is no probability of pupil position when it is completely hidden or, if there is very less probability, when it is almost hidden. For example, with the eyelid half closed or when the person looks up, some part of the pupil is hidden. The method will have to take into account such cases, and the energy is weighted to the appeared pupil area. This method helps to recognize the position even if the eyelid is half closed. When the eyelid is completely closed, the position of the pupil is obviously undefined. It is not a major problem because the eyelid has a great chance to be detected as closed.

We use an easy algorithm to track the eyebrows. An eyebrow is defined as a small region during the initialization, and the position is roughly the center of an eyebrow. It is sometimes difficult to indicate the region of the eyebrow correctly, because some people have very thin eyebrows at both sides. Hence, we use a small region to track the position. To detect the eyebrow shape, first a vertical line goes down from the forehead until the eyebrow is detected. The eyebrow position is given by the maximum self-correlation value of the extracted edge image. After the center of the eyebrow is found, the edge of the brow is followed to the left and right to recognize the shape.

As soon as the pupil and eyebrow locations are detected using the methods described previously, it is possible to guess an eyelid location. When the probability of pupil is large or almost the same as its initial value, this means that the eye is opened. When it is small, the eye may be closed or the person is looking up. The eyebrow position narrows the possible eyelid position. This method helps the detection of the true eyelid position as opposed to a possible wrong detection that may occur with a wrinkle. Thus, the program finds the strongest edge in the considered area and sets it as the eyelid.

After data around the eyes are taken, they are checked again to see if they are in a normal movement compared with templates in the database. For example, if the eyes moved to the opposite direction, the next possible position
of the eyes has to be calculated again. This process improves the robustness and reliability of the whole process.

Until now, we have described facial feature tracking using a video image input. In the subsequent sections, we explain the extraction of animation parameters from speech in real time.

**Phoneme Extraction**

Several approaches for phoneme extraction and voice driven facial animation have been mentioned in the beginning of this paper. This section explains our approach in detail. We take a simpler approach that allows satisfactory results with minimum computation. It should be noted that the resulting speech animation has restrictions as far as high degree of realism is concerned, especially as compared with performance driven facial animation systems, the animation is satisfactory for real-time applications where representation of a clone in his or her own voice is important. Figure 8 shows the speech acquisition and phoneme extraction part of the system. Input speech is sampled at 10 kHz with a frame size of 20 ms. Preprocessing includes pre-emphasis and hamming windowing of the signal. We first explain in brief the principle used for the phoneme extraction and simultaneously describe the implementation in our system.

**Linear Predictive Analysis**

Linear predictive (LP) analysis is a technique widely used for analysis and encoding of speech signal. Wakita [20] concluded that the vocal tract shape variation for sustained vowels is directly related to the linear predictive coefficients. For sustained vowels, the vocal tract shape is uniform along its length without any constrictions and does not change for the duration of the vowel. We calculate 12 coefficients as a result of LP analysis for every frame. We compared and studied the LP coefficients for sustained vowels for various speakers, and a definite pattern observed in these coefficients suggested the use of neural networks for phoneme extraction using the LP coefficients (see Fig. 9). Thus, the problem of recognizing the vowels reduces to a classification problem. A three-layer back-propagation neural network is widely used for a variety of pattern recognition and classification problems [21]. We use the same configuration to classify the LP coefficients. There are 12 input nodes for the coefficients, ten hidden nodes, and five output nodes for the vowels. As a result, one of the five chosen vowels—/a/, /e/, /i/, /o/, /u/—is obtained for every frame. We have chosen these vowels because we notice that the vowels in many languages can be roughly classified into these basic sounds or their combinations/variations. The coefficients for training of the neural network are obtained from the sustained vowel data. The speech data was recorded from 12 male and five female speakers. We ran the training sessions several times on the data studying the classification result. We used five repeated cycles for training, every time using the data in a different random order. Further, we used median filtering to
smooth the resulting recognized vowels. The issues involved in the extraction of some of the consonants are discussed in the subsequent subsections.

**Energy Analysis**

We are aware that the application of the vowels alone for speech animation is not sufficient. The vowel-to-vowel transition and the consonant information, which are very important for realistic speech animation, are missing. The consonants are typically produced by creating a constriction at some place along the length of the vocal tract. During such constrictions or closures, the energy in the speech signal diminishes. Hence, we use the average energy in a speech frame to modulate the recognized vowel. It is calculated as the zeroth autocorrelation coefficient of the frame and has already been computed during the LP analysis phase. Thus, the calculation of energy does not cause any additional computational overhead.

As an initialization process, we record background noise in the room to set the energy threshold for silence. Also, we record sustained vowel /a/ from the user asking her to say the utterance with maximum volume expected in normal speech. This enables us to compute the maximum energy. This value is used to get a normalized weighting factor for the vowels during normal speech.

As explained before, the parameterized face model (MPEG-4 model in our case) enables us to animate these modulated vowels. We have designed a set of the reference FAPs for each of phonemes under consideration. The normalized weighting factor directly proportional to the energy in the speech signal is used to scale the FAPs for the corresponding vowel. Figure 10 pictorially depicts the idea behind modulating vowels with energy envelope.

**Zero Crossing**

Using the energy content of the signal may result in false closure of mouth, especially in the case of affricates and unvoiced fricatives. For such cases, we can use the average zero crossing rate in the speech signal for each frame. The mean short time average zero crossing rate is 49 per 10 msec for unvoiced and 14 per 10 msec for voiced speech [22]. This criterion is useful in making a distinction and is sufficient for our purpose. In case of the presence of low energy in the speech frame, the zero crossing criterion decides the phoneme. Like the vowels, a set of FAPs has been designed for the fricatives and is used as reference for the animation in the subsequent stages.

**Conclusion**

In the previous sections, we have described extraction of FAPs for expressions as well as phonemes. The data extracted in both the steps can be combined and applied to an MPEG-4 compatible face. This way, the stream of output value can be used on any clone, avatar, or even comic characters without any reprogramming. Figure 11 shows...
some animation results for facial tracking whereas Fig. 12 shows the frames of animation “Hello” on a different face model.

This paper has discussed real-time facial animation by facial feature tracking and phoneme extraction, and described in whole the process from the creation of a cloned head to its animation. Effective facial animation of a clone can be achieved using a simple PC connected camera and a microphone. This system works on Windows NT, 2000 and 95/98. The animation speed, including face tracking and lip synchronization, is 10 to 15 frames of animation per second with a Pentium III 500MHz processor and with capturing 15 images per second at 320×240 resolution.

Acknowledgment

The research was supported by European project VPARk (Virtual Amusement Park: ACTS Project No. AC353).

Taro Goto is currently working as a Computer Science Assistant in the MIRALab, University of Geneva. He obtained degrees in electric and information engineering from Gifu University, Japan. After getting the degree, he worked on an image processing and security system. His research interests are man-machine interfaces through natural human motion and 3-D animations created by its input. He is primarily working on research of face/body recognition for creating a 3-D model.

Sumedha Kshirsagar received the bachelor of engineering, instrumentation, from the University of Pune (India), followed by the master of technology in electrical engineering, from the Indian Institute of Technology, Bombay. She is currently working at the MIRALab, University of Geneva, as a Research Assistant and Ph.D. candidate. Her research interests include real-time speech animation, expressive facial animation, emotionally autonomous virtual humans, and communication in virtual worlds. She has contributed to the following European projects during her collaboration at MIRALab: VIDAS, SONG, and INTERFACE.

Nadia Magnenat-Thalmann has researched virtual humans for more than 20 years. She studied psychology, biology, and chemistry at the University of Geneva and obtained her Ph.D. in computer science in 1977. In 1989 she founded MIRALab, an interdisciplinary creative research laboratory at the University of Geneva. Some recent awards for her work include the 1992 Moebius Prize for the best multimedia system, awarded by the European Community; “Best Paper” at the British Computer Graphics Society Congress in 1993; an award from the Brussels Film Academy for her work in virtual worlds in 1993; and election to the Swiss Academy of Technical Sciences in 1997. She is president of the Computer Graphics Society and chair of IFIP Working Group 5.10 in computer graphics and virtual worlds.

References