Synthetic Faces: Analysis and Applications

Sumedha Kshirsagar, Stephane Garchery, Gael Sannier, Nadia Magnenat-Thalmann
MIRALab, CUI, University of Geneva, 24 rue du General Dufour, CH-1211 Geneva, Switzerland

ABSTRACT: Facial animation has been a topic of intensive research for more than three decades. Still, designing realistic facial animations remains to be a challenging task. Several models and tools have been developed so far to automate the design of faces and facial animations synchronized with speech, emotions, and gestures. In this article, we take a brief overview of the existing parameterized facial animation systems. We then turn our attention to facial expression analysis, which we believe is the key to improving realism in animated faces. We report the results of our research regarding the analysis of the facial motion capture data. We use an optical tracking system that extracts the 3D positions of markers attached at specific feature point locations. We capture the movements of these face markers for a talking person. We then form a vector space representation by using the principal component analysis of this data. We call this space "expression and viseme space." As a result, we propose a new parameter space for sculpting facial expressions for synthetic faces. Such a representation not only offers insight into improving realism of animated faces, but also gives a new way of generating convincing speech animation and blending between several expressions. Expressive facial animation finds a variety of applications ranging from virtual environments to entertainment and games. With the advances in Internet technology, the development of online sales assistants, Web navigation aides and Web-based interactive tutors is promising than ever before. We overview the recent advances in the field of facial animation on the Web, with a detailed look at the requirements for Web-based facial animation systems and various applications.

Key words: speech animation; facial animation; expression analysis; principal component analysis; facial animation on the Web; Web-based synthetic characters

1. INTRODUCTION

Animatable synthetic faces have always been of prime importance in computer graphics applications. Facial communication enhances believability, realism, and involvement of the users of such systems. Accordingly, the goal of the facial animation systems has always been toward obtaining a high degree of realism using optimum resolution facial mesh models and effective deformation techniques. In this article we explain the requirements of the facial animation design systems, systematic approaches taken for the design of facial animation, and a few important parameterization schemes used for designing facial animations. We emphasize the importance of parameterization schemes in designing facial animations. We then describe our approach to devise a new parameter space that is effective in creating new speech animations along with expressions blending. Finally, we focus on the facial animation applications on the Web. Irrespective of the techniques used for facial animation design, the Web-based applications pose different requirements regarding real time rendering and visualization of the facial animation over the Internet. We overview such requirements and present a brief survey of various facial animation applications currently found on the Internet.

We start with a detailed explanation of the facial animation design in Section 2. We focus on various effects in facial animation in order to enhance realism. We capture optical tracking data of a real person speaking a number of sentences from a database of phoneme-rich sentences. As a result of the principal component analysis of this captured data, we propose a new parameter space, which is effective for the design of facial expressions as well as for blending various expressions during speech animation. This data analysis and the results have been explained in Section 3. Expressive facial animation finds a variety of applications ranging from virtual environments to entertainment and games. In Section 4, we turn our attention to the most popular application of facial animation, that is, interacting characters on the Web. With the advances in Internet technology, the development of online sales assistants, Web navigation aides, and Web-based interactive tutors is more promising than ever before. We explain basic requirements, evaluation criteria, and some examples of online interactive characters.

2. FACIAL ANIMATION DESIGN: TOP TO BOTTOM

The facial animation design systems typically use the following steps demonstrated in Figure 1.

1. Define a desired face model geometry: male/female, realistic/cartoon, etc.
2. Define an animation structure on the facial model by parameterization.
3. Define "building blocks" or basic units of the animation in terms of these parameters, e.g., static expressions and visemes (visual counterparts of phonemes).
4. Use these building blocks as key frames and define various interpolation and blending functions to generate words and sentences from the visemes and emotions from the expressions. The interpolation and blending functions contribute to the realism for a desired animation effect.
5. Generate the mesh animation from the interpolated or blended key frames.
Out of these, steps 1 and 2 involve input from a developer of the facial animation system, whereas steps 1, 3, 4, and 5 rely heavily on the animator’s or the system user’s input. In some facial animation systems, each key frame may be defined by the complete facial mesh (rather than a set of parameters) to portray a particular expression or viseme. All the vertex positions in the mesh are then interpolated over time to get the final animation. This approach is popularly known as “morph-target” animation and can be used for animating any deformable objects. A parameterized facial animation system is certainly helpful for an animator to design the key frames of animation, rather than working on a low level with the vertices of the mesh. In this section, we first explain the importance of the parametrization schemes for facial animation design. We also explain subsequent steps involved in an expressive speech animation like co-articulation and expression blending.

2.1. Parameterization and Deformation. Optimum resolution mesh models with efficient deformation algorithms form the basic requirements of the facial animation systems: real-time or non-real-time, stand-alone or Web-based. Very high quality animation demands for high-resolution models with carefully designed polygon distribution and more realistic deformation algorithms simulating real life like facial muscle and skin actions. In order to have a high-level control over the facial mesh deformation, a set of parameters is required. These parameters have been often inspired by the facial muscle movements. Such a muscle-based parameter scheme also serves as a guide to implement various deformation models for facial animation. In general, several muscles are involved in generating various facial movements important for speech and expressions. Consequently, more than one muscle-based parameter is required to be manipulated in order to obtain certain facial poses. Figure 2 shows a frontal view of facial muscles.

Parameterization forms an important step for the developer as well as the animator. From a developer’s point of view, the parameterization should take into account the finest control over the face object and the realistic deformations on a low level. From an animator’s point of view, the parameterization should consider a higher level interface for easy, fast, and versatile design. Thus, the goal of the developer of a facial animation design system is to provide the animator with a wide range of natural and intuitive expression control, having subtlety and orthogonality as additional requirements. On one hand, we have low-level parameters that control the actual facial mesh with the finest details simulating all possible movements. This parameter set is dependant on model as well as the technique used for mesh deformation. On the other hand, we have high-level parameters that are easy to use by an animator considering most commonly used facial expressions and visemes. Table I summarizes these important points about facial deformation and parameterization.

One of the earliest parameter schemes, the facial action coding system (FACS) (Friesen, 1978), was developed by Ekman et al. FACS has been widely used as a basic parameter scheme for describing the facial expressions. The system describes the most basic facial muscle actions and their effect on facial expression. All the muscle actions that can be controlled independently and are visually distinguishable are included. Though, originally indented
only for describing facial actions, many other schemes were based on FACS or inspired from FACS for facial animation design. FACS is used as a way to control facial movement by specifying the muscle actions needed to achieve desired expression changes.

The facial animation system developed by Kalra et al. (1992) used the minimal perceptible action (MPA) as basic facial motion parameter. The face deformation was simulated using the rational free form deformation. In comparison with FACS, the MPAs support nonsymmetric movements, nonfacial actions such as “nod head,” “roll head,” and more detailed actions in mouth region, e.g., “pull midlips.”

Recently, the MPEG-4 standard (http://www.cselt.it.mpeg) proposed a scheme for specifying synthetic faces and facial animation using a parameter set. The facial definition parameter (FDP) set and the facial animation parameter (FAP) set are designed to encode facial shape as well as animation of faces, thus reproducing expressions, emotions, and speech pronunciation. The FDPs are defined by the locations of the feature points and are used to customize a given face model to a particular face. They contain 3D feature points such as mouth corners and contours, eye corners, eyebrow centers, etc. Each FAP value is simply the normalized displacement of a particular feature point from its neutral position expressed in terms of the facial animation parameter units (FAPU). The FAPUs correspond to fractions of distances between key facial features (e.g., the distance between the eyes). Thus, once we have the displacements of the feature points from the neutral position, the corresponding values normalized with FAPUs directly give us the FAPs. Figure 3 shows the locations of the feature points as defined by the MPEG-4 standard. For a performance-driven facial animation and analysis of facial motion data (explained in later sections), we use optical markers at the locations of the MPEG-4 feature points.

### 2.2. Complex Effects and Realism

In this section we focus on steps 4 and 5 of the animation design pipeline. Though this is not the main focus of this article, we include this discussion here for completeness and understanding of the facial animation design systems. The steps involve are the definition of key frames and transition from one key frame to the next in a manner that would create a realistic animation. For a rich quality facial animation, a variety of blending and interpolation functions have been considered. These are important especially for speech animation and expression blending, taking into account the effects of co-articulation and smooth transitions from one expression to the other. These blending functions have varied from simple triangular functions for phonemes and attack-sustain-decay-release type of envelopes for expressions to user defined nonlinear interpolation functions. However, these functions have often been designed by the rules based on observation and intuition. Figure 4 shows various typical time envelopes used for the facial animation design. There can be many more possibilities of such interpolation functions.

### Table I. Parameterization and deformation.

<table>
<thead>
<tr>
<th>Parameterization</th>
<th>Deformation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Definition</td>
<td>Generating mesh deformation from a given set of values for the parameter set</td>
</tr>
<tr>
<td>Importance</td>
<td>For developers of FA systems</td>
</tr>
<tr>
<td>Basic requirements</td>
<td>Should fully support selected parameter set, resulting in “realistic” facial movements</td>
</tr>
<tr>
<td>Additional requirements</td>
<td>Speed, ease of implementation</td>
</tr>
</tbody>
</table>

The complexity of the key-frame-based facial animation system increases when we incorporate natural effects such as co-articulation for speech animation and blending between a variety of facial expressions during speech. The use of speech synthesis systems and subsequent application of co-articulation to the available temporized phoneme information is a widely accepted approach (Grandstom, 1999; Hill et al., 1988). Co-articulation is a phenomenon observed during fluent speech, in which facial movements corresponding to one phonetic or visemic segment are influenced by those corresponding to the neighboring segments. Two main approaches taken for co-articulation for computer generated speech animation are by Pelachaud (1991) and Cohen et al. (1993). Both these approaches...
have been based on the classification of the phoneme groups and their observed interaction during speech pronunciation. Pelachaud (1991) arranged the phoneme groups according to the deformability and context dependence in order to decide the influence of the visemes on each other. Muscle contraction and relaxation times were also considered and the facial action units were controlled accordingly. Cohen et al. (1993) defined nonlinear dominance functions for the facial control parameters for each of the visemes and then used a weighted sum to calculate the control parameter trajectories for the co-articulated speech animation. Thus, for each viseme and each facial control parameter, such a dominance function has to be designed.

3. FACIAL MOTION ANALYSIS

In order to produce highly realistic facial animation, advanced techniques such as multiple video cameras, opto-electronic capture, and laser scanners have been used. Contrary to the previously described key-frame approach, in such methods, the movements of the facial feature points are captured for every frame. The animation parameters derived from this captured data are retargeted to various facial models to obtain animation for every frame. It is not always practical to apply such motion capture data for a performance driven facial animation, because it is often restricted by the availability of the performer and complexity of the equipment involved, which often needs tedious calibration and set-up. However, the output of such a motion capture session can be used to design the above-mentioned “building blocks” of the facial animation, thus ensuring an adequate degree of realism at the key-frame level. Furthermore, such motion capture data can be effectively used for analysis of facial movements. In our opinion, the key to improving realism in facial animation is to observe and analyze the actual faces talking and expressing. We analyze such facial motion capture data statistically. As a result of the analysis, we obtain a new parameter scheme for facial expression design, which can also be effectively used for expression blending as explained in this section.

3.1. Data Acquisition. We use a commercially available optical tracking system (VICON 8) to capture the facial movements (http://www.vicon.com). We use the selected MPEG-4 feature points for facial tracking as shown in the Figure 5. We used six cameras for the capture and obtain the 3D trajectories for each marker point as the output of the tracking system. As we focus on speech animation, out of the 27 markers shown in Figure 5, only 8 markers along the outer lip border, 2 markers on the chin, and 4 markers on the cheeks are used for the statistical analysis. The movements of the eyes and eyebrows can be controlled independently from the lip region movement in the facial model and thus are not considered for the analysis. The speaker is made to speak fluently 100 randomly selected sentences from the TIMIT database (http://www.ldc.upenn.edu) of phoneme rich sentences. The head movement of the speaker is not restricted, and thus we need to compensate for the global head movements in order to obtain the local deformations of the markers. For tracking global head movements, three additional markers on a head band are used. We use the improved translation invariant method (Martin and Aggarwal, 1988) to extract rigid movements of the head from the tracked data. Once the global head movements are extracted, the motion trajectories of all the selected feature point markers are compensated for the global movements and the absolute local displacements for each are calculated. A subset of sentences is also captured when the speaker is asked to express different emotions while she spoke. Further, the recordings are also made for static mouth shapes for individual phonemes and six basic facial expressions (joy, anger, sadness, fear, surprise, disgust). For each frame of the captured data, a vector of 3D positions (compensated for global head movements) of the selected 14 markers is extracted.

3.2. Data Analysis. As explained in the previous subsection, for each frame, we have a vector of 42 components (3 coordinates for each of the 14 selected markers). Thus each vector in 3D space is represented as

\[ d = (x_1, y_1, z_1, x_2, y_2, z_2, \ldots, x_{14}, y_{14}, z_{14})^T, \quad n = 14. \] (1)

It can easily be observed that the data are highly interdependent, because of the very nature of the facial movements. For example, the displacements of the points around the lip area are highly correlated to each other, and to the jaw movements, because they cannot be physically moved independent of each other. The lower lip movements are directly linked to the global jaw movements. In addition, there can be local movement of the lips independent of the jaw movement. Similarly movement of the corner lips, as in lip puckering and lip sucking, directly affects the movement of the cheeks.

Figure 5. Placement of markers for optical tracking. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]
Though defining movements of these points are important for generating realistic and precise mesh deformations, the whole set of points is not convenient to use for designing the facial expressions. There are many correlations between the feature point movements, especially for the lip and cheek regions, which make design of facial animation tedious. This especially applies to MPEG-4 FAPs, which are defined by one parameter for each feature point. It is necessary to extract these correlations and come up with a high-level parameter set that is easy to use. However, just observing the capture data in the form of 3D position trajectories does not throw much light on how these movements are interrelated. This interrelation is the key factor for realistic animation, and we use PCA to extract this relation, which occurs because of natural constraints.

3.2.1. Principal Component Analysis. PCA is a well-known multivariate statistical analysis technique aimed at reducing the dimensionality of a dataset, which consists of a large number of interrelated variables, while retaining as much as possible of the variation present in the dataset. This is achieved by transforming the existing dataset into a new set of variables called the principal components (PC). These are uncorrelated and are ordered so that the first few PCs retain the most of the variation present in all of the original dataset. Use of PCA along with facial motion capture data for facial animation has been previously reported (Arsal and Talkin, 1998; Kuratate et al., 1998). However, these approaches were mainly focused on reducing the dimensionality of the facial capture data and further reuse of these data for animation. Our motivation behind using the PCA is to study the correlation between various facial movements and the significance of the principal components for facial animation. The initial results of this analysis and research have been reported in Kshirsagar et al. (2001), Kalberer and Van Gool, 2001 also carried out similar experiments to build speech animations. For a detailed explanation of the PCA, we refer to Jollife (1986). Here we only explain the results of the analysis and their significance in facial animation systems.

We use the entire set of the motion trajectory data of the 3D positions of the selected markers as an input to the PCA analysis. Thus, each input vector is 42 dimensional as explained in Section 3.2. As a result of the PCA on all the frames of the captured data, we obtain a new vector space, whose basis vectors are nothing but the principal components. We call this space as expression and viseme space.

We get the matrix $T$, which forms the transformation matrix between the 3D vector space and the transformed expression and viseme space. Thus, each 42-dimensional vector $d$ corresponding to the 3D position data can be mapped onto a unique vector $e$ in this newly formed PC space:

$$e =Td,$$

where $T$ is an orthogonal matrix and the inverse transformation is given by

$$d = T^Te,$$

where $T^T$ denotes the transpose of the matrix $T$. Each distinct viseme and expression is represented as a point in this transformed multi-dimensional space. The next subsection explains what these “abstract” principal components represent in real life.

3.2.2. Contribution of PCs. Once we obtain the PCs, we can easily establish their role in generating facial animation. We notice that the last three eigenvalues of the covariance matrix of the input data are zero, and thus the dimensionality of the space is directly reduced to 39. In addition, 99% of the variation has been accommodated in only the first 9 principal components. In general, applied to any dataset, the principal components may or may not represent any meaningful parameters. We notice that, for the facial capture data, they are closely related to individual facial movements. To analyze this, we allow only one PC to vary at a time, keeping others at the default neutral position. Then we apply an inverse transformation (Eq. 3) to obtain the 3D position of the markers. From these 3D positions, the MPEG-4 FAPs can be directly extracted by normalization, as explained in Section 2.1. We can use any MPEG-4-compatible facial animation system in order to visualize the results. We use the animation technique described in Kshirsagar et al. (2001).

Figure 6 shows the influence of the first four principal components. They are related to the opening jaw (a), lip protrusion (b), lips opening (c), and vertical movement of lip corners (d), as in smiling. Note that these movements are not controlled locally, meaning that jaw opening does not only result in the vertical displacement of the jaw, but also the rotation movement. Also, the lip opening affects movement of cheeks. Thus, a facial movement obtained by combined activity of bones and associated muscles can be visualized to be controlled by a single PC. The basic movement of opening of jaw, for example, that required handling several parameters using mere MPEG-1 FAPs can be easily controlled by a single parameter. When the PCs are transformed back to FAPs using Eq. 3 above, the FAPs corresponding to lips and jaw rotation as well as cheek movements are appropriately affected. Thus these PCs, in fact, define a new parameter space for facial animation and a more intuitive interface for the designers of the facial animation. These parameters are directly linked to the correlated facial movements. Our observation regarding which PCs typically simulate which

![Figure 6. Contribution of principal components. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.](image)](image)
combined muscle activities is summarized in Table II, with the major muscles affecting mouth region (refer to Fig. 2 for facial muscles).

Though we have used MPEG-4 FAPs as low level parameters, similar analysis and observations could be made with other low level facial animation parameters. It is possible to define a mapping between high-level facial movements and any set of low-level facial animation parameters. This mapping could be by way of a simple grouping and liner or piecewise linear functions or by more complex algorithms. The important point about the mapping we have proposed is that it is derived from the statistical analysis of the data captured from a real person speaking. We observe that the components obtained as a result of the analysis, are directly related to independent facial movements mainly observed during fluent speech.

### 3.3. Speech Animation and Expression Blending

Speech animation typically makes use of the temporized phonemes extracted from either real or synthetic speech as the building blocks. These phonemes are mapped onto visemes for which facial animation parameters are predefined. As mentioned in Section 2, various complex effects such as co-articulation and expression blending add realism to the talking heads. In this section, we address the problem of expressive speech animation by using the expression and viseme space.

We look at an application where synthetic or real speech is used with phoneme segmentation to generate speech animation. For all the phonemes used for speech animation, we transform the 3D position vectors into the newly generated expression and viseme space. These essentially form the key frames of the speech animation, one key frame corresponding to each phoneme in the speech. We then apply co-articulation rules based on Cohen et al. (1993). This leads to the interpolation between key frames after appropriate weighting and overlap between neighboring visemes. The interpolated trajectories for the parameters are then transformed back to the 3D position space, and the FAPs are calculated.

Because the principal components are independent and optimal, it is much easier to design the “dominance functions” for all the visemes. This is attributed to the fact that not only the number of parameters is less, but also the individual parameters are directly related to the independent mouth movements during speech. Figure 7 demonstrates the process for generating speech animation blended with expressions.

The problem of computing appropriate animation parameters arises when combining various facial expressions with speech. Simple addition of the corresponding parameters may not be satisfactory and requires extra care, because the same facial animation parameters or action units may be affected differently by an expression and viseme at the same time. For example, closing of lips is very important for pronunciation of plosives (p, b, etc.), whereas, parting of lips is important during smiling. Thus, while generating a “smiling plosive,” lip closure has to be handled carefully in order to obtain realistic results. We use the expression space for blending.

We capture six basic facial expressions from a performer, namely joy, sadness, anger, fear, surprise, and disgust. The corresponding vectors in the expression space are mapped. When a particular expression needs to be blended with speech animation, the weighted addition of the viseme vector and the expression vector is taken in the expression space. Additional rules are applied to handle special cases like smiling plosives explained above. However, such rules are simpler to write and implement as the principal components independently represent specific mouth movements. The result is transformed back to the 3D position space to calculate the FAPs. Thus, for each viseme vector of the sentence, we add this vector with the expression vector, and the resulting vector is used for the animation, creating the effect of the “expressive viseme.” The addition in the expression and viseme space appropriately scales all the parameters in the 3D position space in a natural and easy way as compared to the scaling of individual facial animation parameters. By enhancing or suppressing only one or two most significant components for a particular expression, expressiveness of the already defined speech animation can be easily controlled.

Figures 8 and 9 show the variation for the principal components after adding expressions to the neutral case. For a happy expression, PC2, that corresponds to the puckering of lips is more significant and affects the resulting animation the most. Similarly, for the sad expression, PC4 is more significant and relates to the vertical movement of the corner of the lips.

So far in this article, we have explained high-level parameter schemes for designing visemes and expressions derived from statistical analysis of facial movements. We have also explained the use of this scheme is designing expressive speech animation. Both these steps form an important part of the content creation for facial animation. Once the content is created, it can be used for a variety of applications, including Web-based facial animation. In the next section, we turn our attention to the applications using interactive virtual faces on the Internet.
4. FACIAL ANIMATION ON THE INTERNET

Until four to five years ago, Web pages had mainly used only the basic elements such as text, images, and audio-video content supported by plain HTML. This resulted in rather static Web pages without any animation and interaction. With the advent of various technologies and languages; Java, Flash, JavaScript, Dynamic HTML, PHP, just to name a few, the Web pages are equipped with multimedia animation and interaction. As Web users become familiar with increased level of interaction and animation, integrating talking virtual characters on Web pages has been the obvious next step.

Why use a real time virtual character and not simply a prerecorded video of a person talking? This sounds like a simple solution. However, one of the most important aspect is the low bandwidth requirement. Some examples presented below use animation parameters to drive facial animation (such as MPEG-4 FAPs) that allow a coding and transmission with a very low bandwidth. These parameters can be decoded on the client web page, in real-time, and rendered on the face model without any transmission overhead. Another very important and interesting advantage of real time virtual character is the interactivity. With a set of videos, a restrictive user environment has to be defined. On the other hand, with a real-time virtual character and a set of tools like Dialogue System and text-to-speech (TTS), we can construct an autonomous system that is able to communicate with the user. Such interactive virtual faces have already found applications in shared virtual environments, entertainment, and games. They are now becoming popular on Web pages as navigation aids, salespersons, hosts, and presenters.

Animated characters on the Web can be divided into 2D and 3D characters. In 2D techniques, a set of predefined images represents all possible animations, which are blended by different ways in order to generate new animations. This technique is computationally less expensive and can be easily integrated into a real-time system. However, the major drawback is that it is impossible to construct on-the-fly animations in real time. Most of the Web sites using 2D characters use predefined animated gif images or similar techniques. A simple text-based interface is available for interaction with the characters. However, the animation is quite monotonous and not well synchronized with the dialogue content. The notable exceptions are the following: http://www.kmpinternet.com uses 2D animation based on Flash plugin and produces audio and lip-synchronized facial animation in response to user text input. Precalculated lip-synchronized facial animation is also displayed by the virtual characters from Winteractive http://www.winteractive.fr. For the reasons mentioned above, we do not include 2D characters on the Web in this study and concentrate only on 3D facial animation.

4.1. Requirements. Section 2 explained the basic steps in facial animation and various techniques used for key-frame animation. For Web-based synthetic characters, the same techniques can be effectively used, with slight modification or adaptation, depending on the applications. For example, morph-target techniques require much less computation time during animation but need a big preprocessing work and data for animation. The muscle-based or geometric deformation techniques are more complex in terms of computation during animation but need only a short preparation stage on the 3D model. Also, constraints on the mesh resolution in such cases may be more stringent, thus making them unsuitable for Web-based applications. Irrespective of the method used for facial deformation, the most important part of the Web-based facial animation systems is the visualization or rendering step. This is significant from the point of view of the user as well as the Web designer who wishes to integrate talking characters on the Web site. For 3D content visualization on the Web, various APIs have been developed. The virtual reality modeling language (VRML) was probably the earliest attempt to enable 3D content on Web pages. Later, various 3D graphics APIs have been developed in order to integrate 3D content on Web pages. Java3D, GL4Java, Cult3D, Shout3D, and Pulse3D are examples of such APIs.

One of the important requirements for 3D content visualization on the Web is installation of a plugin. Most of the APIs, with the exception of Shout3D, require such plugin installation. For common Web users, a plug-less visualization interface is the most preferred one. A plug-in solution, on the other hand, provides the best execution speed and the user should download the plug-in only once (per release).

In addition to 3D content creation and visualization, real-time interaction is more interesting, challenging and more important in modern Web sites. Such interaction provides added value to the already existing content and also improves entertainment levels during Web navigation.

Figure 10 provides an overview of a typical system for virtual character with all the necessary modules. The input can use a

![Figure 8. Generation of “Happy” Speech in “Expression and Viseme” Space](image)

![Figure 9. Generation of “sad” speech in “expression and viseme” space](image)
keyboard and/or a mouse, but also more natural input like the audio and the video. 3D interaction to the mouse events, such as the camera movements, is provided by the Web-based rendering module integrated in the Web page. A higher level intelligent interaction can be provided by an interactivity module. This module is responsible for generating responses to the user input in a natural language or the visual quality models and images are often used on the Web pages. This hampers the visual quality and beauty of the application. There has to be a trade-off between visual quality and size of data.

Fast download: The size of the downloaded data is an important aspect of Web-based animated characters. Quite often, high initial download time is acceptable, but real time interaction should demand as minimum data transfer as possible. Efficient encoding schemes for the animation data and the audio can considerably reduce the data transfer time requirements.

Easy web integration: This issue is important from the Web-page developers point of view. In order to integrate the talking characters in various Web pages, the tools should be easy to use and deliver quick results.

4.2. MIRALab Facial Animation on the Web: A Case Study.
In order to fulfill all the above requirements to have a virtual talking character on the web, we at MIRALab, University of Geneva, have developed a facial animation system that is based on the MPEG-4 facial animation table (FAT) technology and uses the Shout3D rendering engine. Because Shout3D provides a Java API, the visualization is plugless, and the user does not have to install any special software in a Java-enabled browser. With the MPEG-4 FATs technique, good quality models can be animated with minimum real-time computation, and interactive speeds can be easily achieved.

For the development of the Web face applet, the most important and complex work is the FAT construction itself. A FAT is a table specifying the morph-targets for the MPEG-4 facial animation parameters (FAPs) for a given model. The FAPs are optimum for defining reasonable quality facial animations in most cases and easily compressed and transmitted over the network with appropriate encoding. We have developed the tools to automatically build and export FATs. We use the PCs explained in the previous section as high-level parameters for facial animation design, whereas MPEG-4 FAPs are used as low-level parameters for additional refinement and final rendering. To construct FAT, the process consists of applying various intensities of a FAP to the facial mesh and then to make the corresponding vertex deformation table. This process is repeated for each FAP. Additionally, an animator can specify how a particular character speaks (by defining the visemes) and expresses himself using facial expressions. Once the whole set of FATs is designed, the set of the deformation is compiled into the FAT information, used by the Web-based FAT animation engine. Any FAT animation can then be applied to the synthetic face and synchronized with the audio in the Web browser. The details of the system can be found in Garchery and Magnenat-Thalmann (2001). Figure 11 shows the schematic of the design and implementation of the Web-based facial animation applet. This applet can be accessed on the MIRALab website (http://www.miralab.unige.ch) in the “facial animation” section under the “research” topic.

4.3. Web-based Characters: A Survey. Considering the requirements explained in the previous section, we have done a survey of various Web-based characters used on various commercial Web sites for different purposes. The Web-based characters are becoming

Figure 10. Modules for Web-based interactive characters.

Figure 11. Scheme for Web-based facial animation applet. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]
more and more popular. Table III lists the most significant Web sites providing technology and tools for integrating talking characters on the Web. There are many more commercial web-sites that employ these or similar technologies. The applications noted in the table are not necessarily the applications provided by the developers of the technology, but potential applications suggested by them, or implemented by other users of the corresponding technology. As mentioned before, we have included only the 3D characters with some possible interaction in this survey.

5. CONCLUSION
Parameterization and key-frame interpolation form the most important steps in facial animation design systems. In this article, we have discussed both these aspects in the light of conventional techniques and also proposed a new scheme for facial parameterization. Our study and proposal has a sound support of statistical analysis of facial capture data of a real person speaking. Because the correlation between various basic facial movements has been handled by the PCA, the high-level control parameters (PCs) are found to be more intuitive for designing visemes and expressions. These components also provide a powerful tool for generating expressive speech starting from real speech. The high-level components can be easily converted to MPEG-4 FAPs and hence can be used for a variety of compliant facial animation engines and applications. We have developed one such Web-based facial animation system using MPEG-4 FATs and the plugless shout3D rendering engine. We also reviewed other implementations of the talking characters on the Web that have great potential given the existing technology of facial animation and Web visualization.

Quite often, we ask the following question. Do we really need interactive talking characters on the Web sites? We do not have a clear answer yet. As far as the utility is concerned, the answer may be “No.” However, such interactive characters definitely provide a great improvement in the presentation of the content and enhance the enjoyment of the Web experience. Today, the core technology for such talking characters is maturing, and the Web site implementation and integration phase is catching up. More and more Web sites are ready to integrate the technology of talking interactive characters for a variety of applications. We are hopeful that very soon we will have a positive answer to this question. The virtual characters that are being used as added gadgets will definitely be used as services in near future.

ACKNOWLEDGMENTS
We are thankful to the MIRALab staff for their valuable support and help.

REFERENCES
M.M. Cohen and D.W. Massaro, Modelling co-articulation in synthetic visual speech, N. M. Thalmann and D. Thalmann (Editors), Springer-Verlag, New York, 1993, pp. 139–156.