Analysis and Reproduction of Facial Expressions for Realistic Communicating Clones

Stéphane Valente *
Ecole National Supérieure des Mines de Paris
Centre de Robotique
60, Boulevard Saint Michel
E-mail: Stephane.Valente@caor.ensmp.fr

Ana C. Andrés del Valle & Jean-Luc Dugelay
Institut Eurécom,
Dpt of Multimedia Communications,
B.P. 193, 06904 Sophia-Antipolis Cedex, France
E-mail: {ana.andres,dugelay}@eurecom.fr

July 13, 2000

Abstract

This paper presents a novel view-based approach to quantify and reproduce facial expressions, by systematically exploiting the degrees of freedom allowed by a realistic face model. This approach embeds efficient mesh morphing and texture animations to synthesize facial expressions. We suggest using eigenfeatures, built from synthetic images, and designing an estimator to interpret the responses of the eigenfeatures on a facial expression in terms of animation parameters.

Keywords: Face cloning, facial expressions, eigenfeatures, animation, MPEG-4.

1 Introduction

Being able to analyze the facial expressions of a human face in a video sequence and reproduce them on a synthetic head model using a compact set of Face Animation Parameters (FAP) is of tremendous importance for many multimedia applications, like model-based coding, virtual actors, human-machine communication, interactive environments, video-telephony and virtual teleconferencing [1].

*this work is part of his Ph.D. research thesis at the Institut Eurécom

In the literature, three general analysis and animation techniques can be found to perform this task:

(i) **feature-based techniques and animation rules:** these methods are based on parametric face models which are animated by a few parameters directly controlling the properties of facial features, like the mouth aperture and curvature, or the rotation of the eye-balls. The analysis technique consists in measuring some quantities on the user's face, for instance the size of the mouth area, by using blobs, snakes or dot tracking. Some animation rules translate these measurements in terms of animation parameters [2, 3, 4, 5]. These methods have the advantage of being fast but they are not very precise given the rough nature of the analyzed information. Furthermore, most of them require some markers of makeup to highlight the facial features of interest for the analysis algorithms;

(ii) **motion-based techniques and wireframe adaptation:** motion information, computed on the user's face, is interpreted in terms of displacements of the face model wireframe, via a feedback loop. The face model can be either parametric or muscle-based [6, 7, 8]. These techniques have proved to be very precise, especially when a realistic face model is used. However, they are generally slow, as they are iterative: all degrees of freedom in the 3D mesh induced by animation parameters have to be linearized in the image plane to interpret the motion field, and this operation is repeated several times for each analyzed image until some convergence of the FAPs is obtained. Another limitation occurs when the face model has no texture attached to the mesh. In this case, the analysis algorithm is "blind", and has no way to ground the face model to the real face, sometimes failing to track the user;

(iii) **view-based techniques and key-frame interpolation:** face animation is realized by interpolating the wireframe between several predefined configurations (*key-frames*) that represent some extreme facial expressions. The difficulty of this approach is to relate the performer's facial expressions to the key-frames, and find the right interpolation coefficients. This is generally done by view-based techniques, which use appearance models of the distribution of pixel intensities around the facial features to characterize the facial expressions: in [9, 6], template-matching algorithms compute correlation scores with examples found in a database, and interpolation networks (a generalization of neural networks) produce the interpolation coefficients from the correlation scores, whereas [10] directly uses neural networks.

Although view-based techniques and key-frame interpolation are quite intuitive, they suffer from three difficulties implied by the training of the appearance models of the real face. Firstly, the appearance models have to be carefully designed to take into account the coupling between the head pose (the $3D$ position and orientation of the user's face) and the facial expressions. (For instance, if

the performer nods his head downward, his mouth will be curved, and this could be interpreted as a smile). Another way to state this limitation is that the training examples showing the user must be precisely geometrically registered. Secondly, the examples used to train the system must be closely related to the corresponding key-frame; the user must carefully control the intensity of his facial expressions according to the synthetic key-frames during the training phase. This requirement is the most difficult for the user to meet as the face model often lacks realism and cannot be strictly mimicked by a human being. And thirdly, the training database should contain a very large number of real examples and key-frames to cover all degrees of freedom permitted by the synthetic face model. This ideal database requires a user to perform thousands of facial expressions, which is in fact impractical. Needless to say, implementing these analysis and synthesis algorithms with a real person to train a cloning system is a very empiric task.

In this article we describe a new view-based approach where, by using image analysis techniques, we represent facial expressions in terms of animation parameters. To overcome the previously stated limitations for view-based techniques, we suggest using a highly realistic $3D$ head model (i.e. speaker dependent) to generate both the training images and the key-frames [11]. The benefits of replacing a real person by his/her clone during the training phase is that the synthezised database is automatically calibrated, in terms of that geometry and intensity of the facial expressions. The available number of examples and key-frames is also virtually unlimited for sampling the visual space of face expressions across various poses, via the face animation parameters. In Sections 2 and 3 respectively, we explain how our system synthesizes these poses, and how a principal component analysis is performed over the set of training images to extract a small amount of vectors which optimally span the training space. This latter task saves computing time and memory resources for the overall process. The user's facial expression is characterized in the optimally spanned space through a simple correlation mechanism. In Section 4, we deal with decoupling the pose and the expressions in the correlation scores and relating the analysis to the face animation parameters. Since all degrees of freedom permitted by the synthetic face are systematically exploited by the training strategy, our approach is not limited by the amount of available keyframes. Section 5 shows how this is put into practice. Finally, in Section 6 we describe our work within the framework of the MPEG-4 standard.

During the next discussion we assume that we know the precise location of the facial features, and the global head pose: in [12] we presented an efficient analysis-synthesis feedbak loop solving this issue for a video sequence.

2 Synthesis of Facial Expressions

To achieve a higher level of realism for facial expression synthesis we use realistic models and well-designed animation techniques. We build our face models from Cyberware^{TM} range data to obtain a realistic representation of the user [12].

Our models are made of a triangular wireframe, onto which a cylindrical texture is mapped, and they preserve a level of complexity compatible with real-time manipulations. We prefer using a static and unique model and defining its animation for a given person instead of utilizing a generic adapted model which is ready to be animated but is less accurate.

We apply different animation techniques to generate flexible facial expressions [13]. The originality of our approach lies in the use of texture data to synthesize some of the expressions.

- **Mesh morphing:** the animation is obtained by direct manipulations of the mesh vertices (see fig. 1(b)). This technique consists of interpolating the location of the $3D$ primitives between several key-gestures; this can be a straightforward implementation of key-frame animation;

- **Animation of texture coordinates:** we also directly manipulate the texture coordinates associated to the mesh vertices. Unlike mesh morphing, this method animates without physically moving the $3D$ model vertices or altering the shape of the face. This technique can therefore simulate the motion of the face skin over the underlaying bones, and is applied to the eyebrow animation (fig. 1(d));

- **Texture displacements:** we apply different transformations to the cylindrical texture. In most face models, eye movements are synthesized by rotating the two spheres inserted behind the eyelids. We control the gaze direction by drawing the pupils into the texture image. We perform two openings to the original cylindrical texture through its alpha channel (transparency channel) and we insert new textures representing the pupils. By doing so, we can move the pupils without modifying the contour of the eyelids (fig. 1(c));

- **Texture blending:** using the image alpha channel we alter the image texture at rendition time. This technique allows us to easily implement the appearance of wrinkles or face blushing. It is very flexible because the location of the wrinkles can be precisely controlled, instead of embedding them into heavy spline-based meshes. It can easily complement the other techniques by following their animation: on fig. 1(e), the shapes of the forehead wrinkles and the mouth furrows are actually modified by the motion of the left eyebrow (implemented by the interpolation of texture coordinates) and the animation of the mouth (by mesh morphing).

The facial expressions of our model are controlled by face animation parameters (FAP), which can gradually generate a given facial expression [14]. These FAPs are based on the MPEG-4 standard paradigm [15], although they do not directly follow the standard guidelines in terms of compliance (see Section 6).

We define an animation vector,

$$\boldsymbol{V} = (\boldsymbol{P}^T \boldsymbol{\mu}^T)^T = (t_x, t_y, t_z, r_x, r_y, r_z, \mu_1, \cdots, \mu_n)^T$$

(a) (b) (c) (d) (e)

Figure 1: Various animations: (a) neutral face model, resulting from the Cyberware^{TM} acquisition; (b) mesh displacements; (c) texture sliding; (d) texture displacements; (e) texture blending.

which contains the head pose $\boldsymbol{P}$ and facial expression $\boldsymbol{\mu}$ parameters. This compact vector approach, where head pose and facial expressions are jointly represented, allows us to reflect the possible inter-correlation between these differing parameters.

The space of all possible facial expressions and head rotations is sampled by generating a set of $\boldsymbol{V}$ vectors, denoted $\{\boldsymbol{V}_i\}_{i=1,..,m}$, and the corresponding images are synthesized. Image patches for the facial features of interest (like the eyes, eyebrows, and the mouth) are extracted, to produce p datasets of training examples, denoted $\{\{\boldsymbol{D}_{i,j}\}_{i=1,..,m}\}_{j=1,..,p}$ where p is the number of considered facial features.

3 Visual Modeling of Facial Expressions

If we perform an eigendecomposition to reduce the number of images from m to q ($m >> q$) for each of the p datasets $\{\{\boldsymbol{D}_{i,j}\}_{i=1,..,m}\}_{j=1,..,p}$, we will obtain p sets of eigenfeatures $\{\boldsymbol{e}_{i,j}\}_{i=1,\cdots,q}$ which are optimal for decomposing any image of $\{\boldsymbol{D}\}$ with the minimum square error between the image and its reconstruction. That is to say, for any facial expression for a given feature represented by image $\boldsymbol{I}$, there exists a vector $\boldsymbol{\lambda} = (\lambda_1, \cdots, \lambda_q)^T$ such as $\boldsymbol{I} \approx \overline{\boldsymbol{D}} + (e_1|\cdots|e_q)\boldsymbol{\lambda} = \overline{\boldsymbol{D}} + \boldsymbol{E}\boldsymbol{\lambda}$, where $\overline{\boldsymbol{D}}$ is the mean of the corresponding dataset.

The bases $\{\{\boldsymbol{e}_{i,j}\}_{i=1,\cdots,q}\}_{j=1,\cdots,p}$ are ideal for characterizing a new facial expression in the sense that they exploit the visual redundancy of the training datasets to extract some compact and decorrelated parameters to represent facial expressions. As the eigenvectors are constructed for the model facial features, we can refer to them as *eigenfeatures*; they capture the pixel distribution in image patches resulting from both the face pose and facial expression.

To generate the image database from which we extract the eigenfeatures we use the $3D$ head model. Its high realism ensures the correct decomposition of the images obtained from a real face. However, the lighting difference between

the synthetic model at the training stage and the cloned person filmed under unknown conditions at the analysis stage is a difficulty that we have to be aware of. To minimize this difference we include a pre-processing step before the eigendecomposition is performed.

At present we are using optical flow based pre-processing techniques. The first one compares the image of the current facial feature with the image of the face in a neutral state. We compare with a reference image to be coherent with the training process, because the number of eigenfeatures in a database generated from optical flow data obtained from frame to frame would be too large. Our approach is somewhat unstable with regard to lighting changes during the analysis because it is not feasible to ensure the same lighting conditions from the first image (face in its neutral position) to the images obtained from frames several seconds away. We are improving this technique by studying the optical flow information from previously analysized frames, without penalizing the training step. We are also including the information obtained from the optical flow between adjacent frames. Other pre-processings, like the use of normalized images image gradients or normalized color images are also candidates as possible improvements for our approach.

4 Analysis and Reproduction of Facial Expressions

Once the facial expressions are visually modeled by the previous eigendecomposition, a facial expression performed by the user, represented by image $\boldsymbol{I}$, is processed as follows (assuming the head pose has already been estimated): the facial features are correlated with the p bases of eigenfeatures, leading to the scores $\{\{\lambda_{i,j}\}_{i=1,\cdots,q}\}_{j=1,\cdots,p}$, which, together with the head rotation parameters, form the vector

$$\boldsymbol{\lambda} = (r_x, r_y, r_z, \lambda_{1,1}, \lambda_{2,1}, \cdots, \lambda_{q-1,p}, \lambda_{q,p})^T$$

Once again we include the rotation information along with the expression components $\lambda_{i,j}$ to denote the possible inter-correlation between these two kinds of parameters. This proves to be of great help when trying to relate the vector $\boldsymbol{\lambda}$ to some vector $\boldsymbol{\mu}$ while decoupling the head pose from the facial expression. To establish such a relationship we have envisaged two possible estimators that are built during the training process:

(i) **Linear estimator:**
We construct the linear estimator $\boldsymbol{L}$, which best satisfies the relation $\boldsymbol{\mu} = \boldsymbol{L}.\boldsymbol{\lambda}$ on the training database in the least mean square sense. One can readily verify that this linear estimator is given by

$$\boldsymbol{L} = \boldsymbol{M}\boldsymbol{\Lambda}^T(\boldsymbol{\Lambda}\boldsymbol{\Lambda}^T)^{-1}$$

where $\boldsymbol{M} = (\boldsymbol{\mu}_1|\cdots|\boldsymbol{\mu}_d)$ and $\boldsymbol{\Lambda} = (\boldsymbol{\lambda}_1|\cdots|\boldsymbol{\lambda}_d)$ are the matrices obtained by concatenating all $\boldsymbol{\mu}$ and $\boldsymbol{\lambda}$ vectors from the training dataset.

(ii) **Radial Basis Function (RBF) based:**
RBF networks were primarily investigated in the literature to appraximate multidimensional surfaces[16]. In our application, they are used to model the relationship between the $\boldsymbol{\lambda}$ vectors, which are observed on the synthetic face, and a given FAP in the training database. Within our analysis framework, one RBF network is built per FAP, that is one per each of the components of $\boldsymbol{\mu}$:

$$\boldsymbol{\mu_i} = \sum_{i=1}^{N} c_i h_i \left(G_i \left(\boldsymbol{\lambda}, \boldsymbol{\lambda}_i \right) \right). \tag{1}$$

The relationship is modeled in equation (1) by adding the contributions of real-valued functions $h_i \left(G_i \left(. \right) \right)$ based on the distance between $\boldsymbol{\lambda}$ and each example vector $\boldsymbol{\lambda}_i$ in the training database (hence the name of Radial Basis Function, where $\boldsymbol{\lambda}_i$ acts as the center of the radial distance symilarity). Its matrix representation is

$$\boldsymbol{H}\,\boldsymbol{C} = \boldsymbol{M}. \tag{2}$$

where,

$$\boldsymbol{M} = \left(\mu_1, \cdots, \mu_N \right)^T$$

and

$$\boldsymbol{H} = \begin{pmatrix} h_1(G_1(\boldsymbol{\lambda}_1)) & \cdots & h_N(G_N(\boldsymbol{\lambda}_1)) \\ \vdots & \ddots & \vdots \\ h_1(G_1(\boldsymbol{\lambda}_N)) & \cdots & h_N(G_N(\boldsymbol{\lambda}_N)) \end{pmatrix}.$$

During the training process we determine the coefficients $\boldsymbol{C} = \left(c_1, \cdots, c_N \right)^T$ by solving (2) using Least Square Inversion,

$$\boldsymbol{C} = \left(\boldsymbol{H}^T \boldsymbol{H} \right)^{-1} \boldsymbol{H}^T \boldsymbol{M}. \tag{3}$$

Our RBF network takes the correlation (4) and the normalized correlation (5) between the image being analysed and the training eigenfeatures as a likeness measurement. We apply the interpolation function h(r)=r,

$$\mu = \sum_{i=1}^{N} c_i \left(\sum_{j=1}^{r} \lambda_j . \lambda_{j,i} \right), \tag{4}$$

$$\mu = \sum_{i=1}^{N} c_i \frac{\left(\sum_{j=1}^{r} \lambda_j . \lambda_{j,i}\right)}{\sqrt{\sum_{j=1}^{r} \lambda_j^2}}. \quad (5)$$

Logically, the denominator of Eq. (5) should be

$$\sqrt{\sum_{j=1}^{r} \lambda_j^2} . \sqrt{\sum_{j=1}^{r} \lambda_{j,i}^2},$$

but since the energy of $\boldsymbol{\lambda}_i$ is constant, this term is automatically compensated by the estimation of c_i.

It can be shown [17] that physically, the correlation and normalized correlation of $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda}_i$ vectors correspond to the correlation between the analyzed image and the training image approximated by $\boldsymbol{\lambda}_i$, in a computationally efficient manner.

The main advantage of our highly realistic approach is that during the training process, when the estimator is defined, we do not require speaker participation. Its $3D$ head model is used to generate the video images which finally train the complete system. The use of a realistic clone ensures a correct correspondance between analysis and synthesis ($\boldsymbol{\mu}$, $\boldsymbol{\lambda}$). The main drawback is that although $\boldsymbol{\mu}$ is speaker independent (FAPs can be synthesized in any clone no matter how we have obtained them), the estimator is completely speaker dependent; the use of the speaker model during the training forces the system to define a dedicated estimator and database per user.

5 System Evaluation

We first experimented our approach on synthetic images to validate the analysis framework. Training datasets were created using the following simple sampling strategy: each FAP was altered one by one, taking the values $\{-1, -0.5, 0, 0.5, 1\}$, and for each obtained facial expression, we synthesized the face model under 9 different orientations, by setting the X and Y rotations to $\{-3, 0, 3\}$ degrees. We tested our pre-processing techniques and estimators on several sets of eigenfeatures. The size of the sets varied depending on the amount of energy stored from the initial database.

The results we obtained showed that the linear estimator and the RBF estimators have similar performance in equal conditions (same percentage of energy and same pre-processing). Once the number of eigenfeatures in our dataset reaches a certain level of energy (99%, corresponding to 22 eigenfeatures for the eyes out of 729 images [17]) all estimators give good results; they estimate the

FAP with an accuracy of around 5%. Pre-processings (for the feature extraction in real images) are difficult to compare because they capture energy differently and they lead to linear systems that have different degrees of freedom. The optical flow pre-processing is comparable to others in terms of size order although it needs more computing.

Figure 2 show the resynthesis of some facial expressions using the linear estimator. The analysis-synthesis approach works quite well for the animation of the eyes and eyebrows, and for expressions that are close to the training dataset in general, suggesting that the analysis strategy makes sense. However, it may have difficulties for some complicated expressions of the mouth, because many FAPs interact altogether in this area of the face, and the obtained facial expressions are too far from the training dataset, which is too simple for complex expressions.

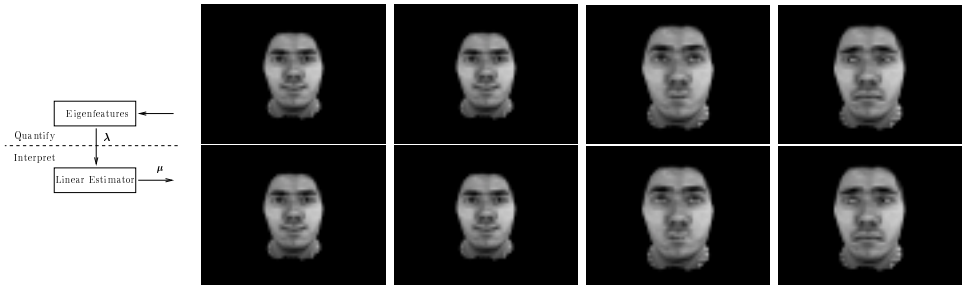

Figure 2: Some analyses of facial expressions: each image of the upper row (which does not belong to the training dataset) was quantified by some eigenfeatures, giving a $\boldsymbol{\lambda}$ vector. A linear estimator mapped $\boldsymbol{\lambda}$ to the animation parameters $\boldsymbol{\mu}$, which are rendered into the images of the lower row.

Some other tests were carried out over real images (see Figure 3). It is difficult to determine which estimator and pre-processing technique gives the best results for images obtained under a different illumination than the training set because the different techniques do not equally influence all FAPs. Nevertheless, the linear estimator could be considered the best due to its simplicity. And, although the gradient pre-processing is slightly more appropriate over the synthetic data, we prefer the optical flow technique because it can be greatly improved to become more lighting independant: in our current implementation, the optical flow computation is carried out with a basic algorithm, which is expected to be more sophisticated in future experiments.

The use of the clone during the complete process proved to be a great advantage. The clone gives more flexibility and freedom for building the complete system. Nevertheless our system continues to be speaker dependent and training the system by synthesizing expressions with the clone prevents us from knowing exact speaker behavior.

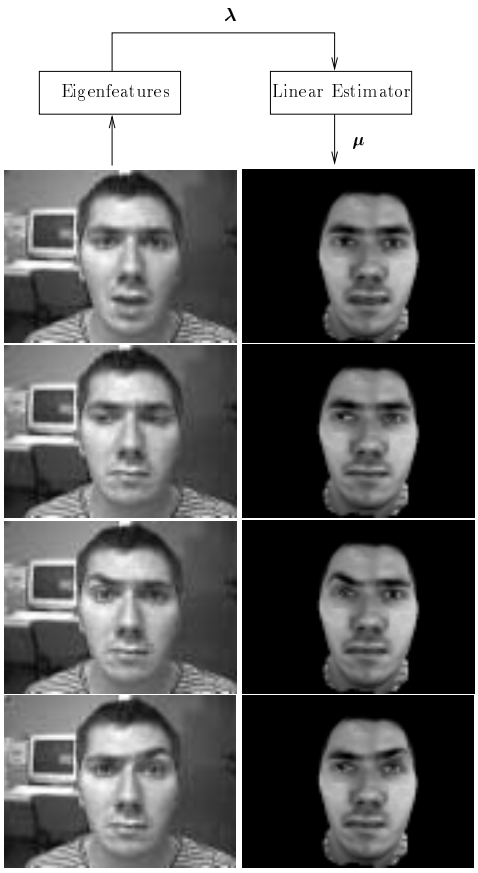

Figure 3: Some facial expression analysis performed over real images

6 The analysis-synthesis scheme under MPEG-4 compliance

Our system has been designed to ensure the maximum synthesis cooperation during face expression analysis, but our animation parameters $\boldsymbol{\mu}$ have been designed without following any guidelines so far.

The main reason for developing communicating clones is to decrease the data to be transported over the network and to offer $3D$ facilities (change of scene, free displacement of the model inside the scene, etc.). With clone communication, video frame data is substituted by sequences of FAPs. The scheme of communication is as follows (Fig. 4): at one end the system analyzes the expressions of the user, translates them into FAPs and encodes them. Then, these FAPs are streamed over the network and finally, a decoder synthesizes the remote clone with the incoming FAPs. MPEG-4 specifies the decoding of FAP

streams and several visual requirements for the face object to ensure the proper interpretation of FAPs in all decoders.

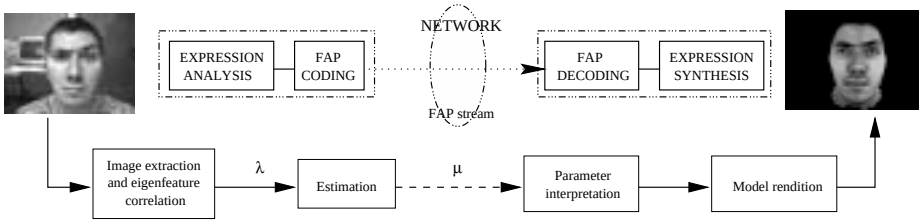

Figure 4: Face animation diagram. The user is recorded by a camera, and an expression analysis system generates the face animation parameters (FAP, μ) which are sent over the network. A decoder interprets them and synthesizes the expressions on the clone.

If we want our FAPs to be become MPEG-4 compliant there are two possible paths to follow. On the one hand, we could define a correspondance $\boldsymbol{S}$ between $\boldsymbol{\mu}$ and $\boldsymbol{\mu}'$, where $\boldsymbol{\mu}'$ will comply with the norm,

$$\boldsymbol{\lambda} \stackrel{\boldsymbol{L}}{\rightarrow} \boldsymbol{\mu} \stackrel{\boldsymbol{S}}{\rightarrow} \boldsymbol{\mu}'.$$

This correspondance would decompose our FAPs, which are currently more complete than MPEG-4's FAPs, in smaller units. On the other hand, we could redefine our FAPs so they are MPEG-4 compliant; this way we would directly synthesize $\boldsymbol{\mu}'$. In such a case, the complete system would have to be retrained and a new estimator, $\boldsymbol{L}$', would have to be built,

$$\boldsymbol{\lambda} \stackrel{\boldsymbol{L}'}{\rightarrow} \boldsymbol{\mu}'.$$

In the rest of this section, we evaluate the nature of our clones and our synthesis techniques in terms of MPEG-4 compliance:

(a) **Geometric modeling:** MPEG-4 does not provide a specific 3D model to be used. It only specifies a face model in its neutral state, a number of vertices on it as reference points (Face Description Parameters – FDP), and a set of FAPs. It also provides the means to tell a decoder how a face object should be animated with Face Animation Tables (FAT) and Face Interpolation Tables (FIT). MPEG-4 specifies 84 FDPs located in specific vertices of the head mesh. FAPs animate taking as reference those positions and the FAP Units (FAPU). FAPUs are distances between certain vertices of the 3D model in its neutral state.

Due to these requirements, our head models are not yet MPEG-4 compliant. Despite this, the great number of primitives the models have makes it relatively easy to arrange the vertices so they comply with the FDPs; FAPUs can be directly computed from the FDPs.

(b) **Synthesis of expressions:** MPEG-4 conceives animation of expressions as a combination of vertix movements (mesh morphing) although there is no obligation to use this kind of animation. Using other techniques (texture displacements, interpolation of texture coordinates or texture blending) would restrict us to proprietary MPEG-4 decoders that are capable of such animations.

For instance, we are using the texture displacement technique to implement the model teeth and tongue by overlapping several texture portions on a plane just behind the model lips. This procedure is not suitable for articulating the tongue out of the mouth in the way that MPEG-4 allows. On the other hand, this solution has the advantage of being more realistic than using generic primitives as it takes the teeth and the tongue of an individual into account.

To use our synthesis techniques and still be MPEG-4 compliant, the system should be able to translate all techniques in terms of mesh movements and to interprete mesh movements, characterized by FATs, in terms of texture displacements. A well-established bijective relationship would allow our head models to be used by other decoders. It would also let our system animate unknown clones. The use of these synthesis techniques for analyzing expressions does not prevent the system from generating completely compatible FAPs. In fact, the FAP encoding and decoding is independent of the expression analysis.

An MPEG-4 terminal that is able to decode FAP streams can animate its own proprietary model. To use non-proprietary head models the decoder interprets an unknown model and its animation behavior before starting the movement synthesis. The models and the animation rules should be defined following MPEG-4 syntax which is based on VRML. Our head models are VRML compliant and we consider that it should be straightforward to build an MPEG-4 scene graph from them.

7 Conclusions

A wide range of applications such as teleconferencing, e-commerce, mobile video communications, etc. will enrich their human-computer interface by integrating realistic clones. The use of realistic clones allows us to build a new algorithmic framework based on image analysis-synthesis techniques. A view-based approach to quantify and reproduce facial expressions on a synthetic head model, by systematically exploiting the degrees of freedom allowed by a realistic face model proves to be a good alternative for obtaining face animation. We propose to use eigenfeatures, built from synthetic images, and to design an estimator to interpret the responses of the eigenfeatures on a facial expression in terms of animation parameters.

Although there are many issues to be resolved, this article contributes to the reproduction of realistic facial expressions by 3D face models. Our auto-

matic framework has the great advantage of giving highly realistic results, and, above all, is able to interpret reality and translate it in a set of compact animation parameters while using efficient non-iterative analysis algorithms. Our current work focuses on the improvement of the pre-processings necessary to overcome the changes of lighting between the training and analysis stages. In the near future, it could become a viable alternative to real–time avatar animation techniques, generally animated from textual commands or video input with markers.

Our face analysis-synthesis techniques are suitable for integration into an MPEG-4 face animation system. Our algorithms can be easily used following the standard. A complete integration will provide a well defined means for realistic clone communication. It will also allow our system to interact with other different animation systems.

Acknowledgements
This work is partially supported by FRANCE TELECOM Research.

References

[1] J.-L. Dugelay, K. Fintzel, and S. Valente. Synthetic/natural hybrid video processings for virtual teleconferencing systems. In *Picture Coding Symposium*, Portland, Oregon, April 1999.

[2] D. Terzopoulos. Modeling living systems for computer vision. In 14^{th} *International Joint Conference on Artificial Intelligence*, pages 1003–1013, Montreal, Quebec, August 1995.

[3] T. S. Huang, S. C. Reddy, and K. Aizawa. Human facial motion modeling, analysis and synthesis for video compression. *SPIE Vol. 1605 Visual Communications and Image Processing'91: Visual Communication*, pages 234–241, 1991.

[4] A. Saulnier, M.-L. Viaud, and D. Geldreich. Real–time facial analysis and synthesis chain. In *International Workshop on Automatic Face— and Gesture— Recognition*, pages 86–91, Zurich, Switzerland, 1995.

[5] J. Ohya, Y. Kitamura, F. Kishino, and N. Terashima. Virtual space teleconferencing: Real-time reproduction of tridimensional human images. *Journal of Visual Communication and Image Representation*, 6(1):1–25, March 1995.

[6] I. A. Essa, S. Basu, T. Darrell, and A. Pentland. Modeling, tracking, and interactive animation of faces and heads using input from video. In *Computer Animation'96 Conference*, Geneva, Switzerland, June 1996.

[7] P. Eisert and B. Girod. Facial expression analysis for model–based coding of video sequences. In *Picture Coding Symposium*, pages 33–38, Berlin, Germany, September 1997.

[8] H. Li, P. Roivainen, and R. Forchheimer. 3–D motion estimation in model–based facial image coding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):545–555, June 1993.

[9] T. Darrell, I. Essa, and A. Pentland. Correlation and interpolation networks for real–time expression analysis/synthesis. Technical Report 284, MIT Media Lab.

[10] F. Hara and H. Kobayashi. An animated face robot — Its component technology development for interactive communication with humans. In *L'Interface des Mondes Réels & Virtuels*, Montpellier, France, Mai 1996.

[11] S. Valente and J.-L. Dugelay. Analysis and reproduction of facial expressions for communicating clones. In *IEEE Workshop on Multimedia Signal Processing*, Copenhagen, Denmark, September 13-15 1999.

[12] S. Valente and J.-L. Dugelay. A visual analysis/synthesis feedback loop for accurate face tracking. *Signal Processing: Image Communication*, to be published in 2000.

[13] S. Valente and J.-L. Dugelay. Face tracking and realistic animations for telecommunicant clones. *IEEE multimedia*, Spring 2000.

[14] MPEG demo of the animation system. URL http://www.eurecom.fr/~image/TRAIVI/animation.mpg .

[15] Information technology — coding of audio–visual objects: Visual — ISO/IEC 14496–2. Comittee Draft ISO/IEC JTC1/SC29/WG11 N1902, International Organisation for Standardisation, Fribourg, Switzerland, October 1997.

[16] Will Light, editor. *Advances in Numerical Analysis: Wavelets, Subdivision Algorithms, and Radial Basis Functions*. 1992.

[17] S. Valente. *Analyse, Synthèse at Animation de Clones dans un Contexte de Téléréunion Virtuelle*. PhD thesis, Ecole Polytechnique Fédérale de Lausanne, Institut Eurécom, France, 1999.

Stéphane Valente received his PhD in communications systems from the Swiss Federal Institute of Technology (EPFL), Lausanne, in 1999, and the French State degree of Ingénieur des Télécommunications (telecommunications engineer) from the National Institute of Telecommunications, Evry, France, in 1994. His research background includes an eight-month internship in speech recognition in the Speech Technology Laboratory (a subsidiary of Panasonic) in Santa Barbara, California, and a 12-month position in the Scientific Data Visualization Department at the Commisariat à l'Energie Atomique (CEA, or Atomic Energy Agency) in Bordeaux, France. His research interests are image and video processing, image synthesis, $3D$ analysis, and virtual reality. In January 2000, he joined the Ecole des Mines de Paris (Paris School of Mines) as a research engineer.

Ana C. Andrés del Valle received the Spanish State degree of Ingeniería Superior de Telecomunicaciones (telecommunications engineering) from the Escola Tècnica Superior d'Enginyeria de Telecomunicacions de Barcelona (ETSETB, or Barcelona Technical School of Telecommunications) at the Polytechnic University of Catalonia (UPC), Barcelona, Spain. From February to November 1999, she was an intern in AT&T Labs - Research, New Jersey, working on face animation. At present, she is a PhD student in the Multimedia Communications Department at the Eurécom Institute, Sophia Antipolis, France. Her research interests are image and video processing, specially face expression analysis, virtual reality and computer graphics.

Jean-Luc Dugelay is a research associate in the Multimedia Communications Department at the Eurécom Institute, Sophia Antipolis, France. He received a PhD in computer science in 1992 from the University of Rennes, France. From 1989 to 1992, he worked at the Centre Commun d'Etudes de Télédiffusion et Télécommunications (CCETT, or Joint Center for Broadcasting and Telecommunications Studies)

in Rennes, for the Centre National d'Etudes des Télécommunications (CNET, or National Center for Telecommunications Studies)/France Telecom R&D Center where he was involved in research on $3D$ motion and stereoscopic television. His research interests include image processing and coding, watermarking and indexing, video communications, virtual reality, and $3D$ imaging. He is a member of the IEEE Signal Processing Society Multimedia Signal Processing Technical Commitee (IEEE MMSP TC), a member of the editorial board of the journal *Multimedia tools and Applications* and an Associate Editor of the IEEE *Transactions on Image processing* (2000-).