appears in: **IEEE International Conference on Multimedia Computing and Systems (ICMCS 99), June 7-11, 1999, Florence, Italy**

Face Tracking and Realistic Animations for Telecommunicant Clones

Stéphane Valente & Jean–Luc Dugelay

Institut Eurécom, Multimedia Communications Department
B.P. 193, 06904 Sophia–Antipolis Cedex, France
{valente, dugelay}@eurecom.fr

Abstract

We present recent developments in the context of face cloning using view–based techniques, permitted by the cooperation between analysis and synthesis algorithms.

We detail efficient $3D$ face modelling techniques, that are capable to reproduce convincing facial expressions, and we show how such realistic face models are suitable for a view–based analysis framework, to recover both the global pose and the facial expressions of a human performer, without any markers or makeup.

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 is of tremendous importance for many multimedia applications, like model–based coding, virtual actors, human–machine communication, interactive environments, video–telephony and virtual teleconferencing.

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, using blobs, snakes or dot tracking. Some animation rules translate the measurements in terms of parametric animation parameters [18, 8, 15]; Some algorithms are already meeting real–time analysis frame rates, while allowing the performer some degrees of freedom in his head position and orientation [17, 1].

(ii) **motion–based techniques and wireframe adaptation:** the 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, 5, 10]; it is necessary to note that these techniques can also handle the task of determining the head pose, by combining global motion information in the regularization algorithms. However, such algorithms are far from achieving real–time performance.

(iii) **view–based techniques and key–frame interpolation:** the face animation is realized by interpolating the wireframe between several predefined configurations (*key–frames*), representing 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 [3, 6], template–matching algorithms compute correlation scores with examples found in a database, and interpolation networks (a specific class of neural networks) produce the interpolation coefficients from the correlation scores, whereas [7] directly uses neural networks to estimate facial expressions from image pixels.

Although view–based techniques and key–frame interpolation are quite intuitive and the prefered methods for real–time performance, they suffer from two difficulties. 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 it could be falsely interpreted as a smile), and the examples used to train the system must be closely related to the corresponding key–frames. This mainly is the reason why such algorithms generally require the performer to stay in a strict frontal view in front of the acquisition camera. And secondly, such a system is limited by the number of key–frames and training examples, as it will be difficult to analyze and reproduce a facial expression that cannot be mimicked by a linear combination of key–frames. Needless to say, implementing these analysis and synthesis algorithms, by matching "by hand" the view–based examples with the corresponding synthesis parameters, is a very empiric and inaccurate task.

2. Overview of our Work

This paper presents our face cloning research in the TRAIVI[1] project [4], which focuses on *Virtualized Reality*, as opposed to *Virtual Reality*: the concept of virtualized reality was developped by Kanade *et al* in [9], pointing out that the world fine details should be taken into consideration in virtual worlds, rather than unrealistic CAD models, to represent the participants or their environment. To enforce a high level of realism, we consequently propose to use person–dependent textured face models built from CYBERWARE^{TM} range data [2].

We noticed that in the literature, unfortunately, none of the face cloning approaches takes advantage of the visual realism of their face model to track and/or analyze facial deformations. Therefore, we offer a novel approach in face cloning, using a *visual* feedback loop, making the analysis and synthesis modules cooperate, both to recover the user's position and orientation, and his facial expressions. To the best of our knowledge, our framework is the first attempt to implement such a cooperation for view–based techniques. In section 3, we review a global motion tracking algorithm that can precisely recover the head pose of the user. The main contribution of this paper resides in section 4, where we transform a static speaker–dependent textured wireframe into an animated face model capable of realistic facial expressions, manipulable by a generic analysis/synthesis framework, thanks to efficient and original deformation techniques. And finally, we will introduce in section 5 how we intend to analyse facial expressions using our animated face models in a view–based manner.

[1] *TRAitement des Images VIrtuelles* (Processing of Virtual Images)

3. Head Pose Determination

We wrote a face tracking and pose determination system which proceedes as follows (figure 1):

Initialization:

(i) the user aligns his head with his head model, or alternatively modifies the initial pose parameters to align his head model with his head;

(ii) when done, an $3D$ illumination compensation algorithm is run, to estimate the lighting parameters that will reduce the photometric differences between the synthetic face model and the real face in the user's environment;

Main Loop:

(i) a Kalman filter predicts the head $3D$ position and orientation for time t;

(ii) the synthetic face model generates an approximation of the way the real face will appear in the video frame at time t; this approximation includes both geometric distortions, scales and shaded lighting due to the speaker's pose, as well as some clues about the location of the background with respect to the face tracked regions;

(iii) patterns are extracted from the synthesized image, representing contrasted facial features (like the eyes, eyebrows, mouth corners, nostrils);

(iv) a differential block–matching algorithm matches these patterns with the user's facial features in the real video frame;

(v) the $2D$ coordinates of the found positions are given to the Kalman filter, which estimates the current head $3D$ position and orientation.

The strength of the visual feedback loop is that it implicitely takes into account the changes of scale, geometry, lighting and background with almost no overload for the feature–matching algorithm: due to the synthesis module that performs a $3D$ illumination compensation scheme, the synthesized patterns will predict the geometric deformations, the lighting and the background location of the user's facial features, making the differential block–matching stage more robust. This enhanced analysis/synthesis cooperation results in a stable face tracking framework without artificial marks highlighting the facial features, supports very large rotations out of the image plane (see figure 2), and even copes with low–contrasting lightings (see the MPEG demo [13]). More details can be found in [19].

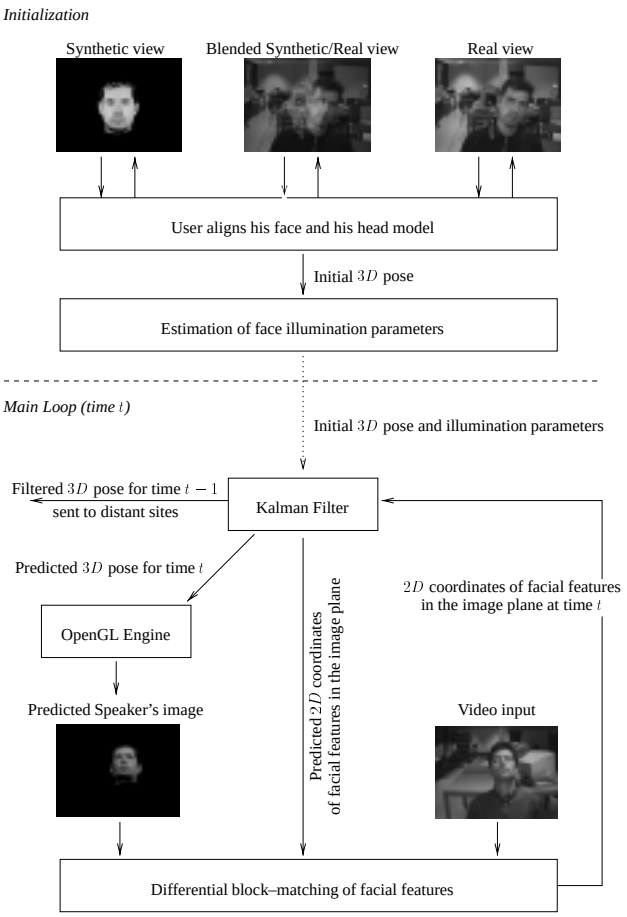

Figure 1. System overview: initialization, and face tracking loop

Figure 2. Head rotations supported by the face tracking system, without any marker or specific lighting

4. Synthesis of Facial Expressions

Most studies about $3D$ face model construction in the literature try to adapt a more–or–less generic face model to an individual from photographs or range data. The construction of facial animations is then trivial, because they are directly built in the generic head model by defining wireframe deformations. As always, there is a tradeoff between real–time rendition capabilities and realism, and the face model may end up being an oversimplified unrealistic avatar. Instead of starting from a generic face model to make it specific to a given person, we took the opposite approach, starting from person–dependent data (range and texture image) corresponding to a neutral facial expression, and processing it to make it manipulable by a generic analysis/synthesis framework. The main difficulty is that, though highly realistic, our face model comes unanimated: it is made of static vertices (with refinments around the facial features to improve the modelling precision), attached to a static texture image via static texture coordinates. Another major difficulty is that there are no separate primitives for the eyeballs: the initial face model is just a plain surface. Nevertheless, this section will show how facial expressions can be achieved by applying simple deformations, not only on the wireframe vertices, but at the three different levels (vertices, texture coordinates, and texture image), by implementing well–known or original animation techniques.

The next figures will display two points of view of the same model in different facial expressions, to emphasize that our animation methods are valid in $3D$, as required by a virtual teleconferencing system. For future comparisons, the initial face model in a neutral facial expression is given in figure 3.

Figure 3. Katia's original face model

4.1. Mesh Animations

Key–frame animation (or *mesh morphing*) consists in interpolating the positions of the mesh vertices between extreme facial expressions. It is particularly suitable for real–time and performance animation, because it involves only linear combinations between predefined vertex positions, and allows to smoothly deform a surface as complex and pliable as the human face. It generally produces less unwanted effects like bulging, creasing and tearing than does facial animation created with bones or lattices [11]. The only requirement for this technique is to have a collection of separate wireframes in different expressions with the same number of vertices in the same exact order, which can be obtained by editing the original wireframe [16]. This animation technique is implemented in our face models to animate the eyelids or the mouth (figure 4).

Figure 4. Mesh vertices displacements: right eye is closed, left eye is half–closed, and the mouth is squeezed

4.2. Texture Coordinates Displacements

All animations do not require to deform the shape of the face model. For instance, lifting an eyebrow corresponds to a shift of the underneath muscles onto the face skull. We mimicked this operation by extending the principle of key–frame mesh interpolation to the texture coordinates, to make the texture image slide over the wireframe.

Figure 5. Texture coordinates displacements: left eyebrow is down, the right one is up

Figure 5 shows that this technique can correctly implement the motion of the eyebrows, and simulate the extension of the skin just below the right eyebrow, by pulling up the eye's makeup, while keeping the head shape unaltered. Such an effect would be impossible to achieve by mesh morphing alone.

4.3. Texture Animations: Texture Displacements

The cylindrical texture mapped onto the mesh vertices can be altered at rendition time to produce animations. In most face models, the gaze is controlled by rotating some eye–balls appearing through holes created in the wireframe for the eyes. Instead of adding new primitives for each eye, we created holes in the texture image (via the transparency channel), and two separate textures behind the main one can be displaced to alter the model's gaze direction (figure 6). Due to the alpha channel, the shape of the eye contours remains unchanged, and covers the moving texture portions. We are also using this technique to implement the model's teeth on tongue, by means of overlapping several texture portions on a plane just behind the model's lips.

(a) (b)

(c)

Figure 6. Texture–shifting in the texture plane: (a) neutral position of the eyes in the cylindrical texture; (b) left shift of the eyes; (c) result after $3D$ mapping

4.4. Texture Animations: Texture Blending

Beside moving some texture portions, it is possible to blend several textures together to produce a new one, for example to fade wrinkles into the model texture at low–cost in

terms of real–time animation, instead of hard–coding them in heavy spline–based meshes [20], as seen on figure 7.

Figure 7. Expression wrinkles and furrows by texture blending

4.5. Realistic Animations

Each defined model alteration is controlled by a single parameter μ_i, a FAP (*face animation parameter*, conforming the guidelines of the MPEG–4 standard [14]). Combining n parameters (i.e. n independent mesh or texture modifications) in a single vector $\boldsymbol{\mu} = (\mu_1, \cdots, \mu_n)^T$, the face model is then capable of complex facial expressions. Although the construction of the deformations is highly person–dependent, the facial expressions are simply controlled by the $\boldsymbol{\mu}$ vector, which is completely transparent for the analysis and synthesis frameworks (figure 8).

If the $\boldsymbol{\mu}$ vector is related to the same FAPs across different face models, although each FAP is implemented in a strict person–dependent manner, it will be possible to analyse the facial expressions of a performer using his own model (like in figure 1), and reproduce them on another $3D$ model. Interested readers are invited to download the MPEG sequence [12] to see several face models interpreting the same $\boldsymbol{\mu}$ parameters.

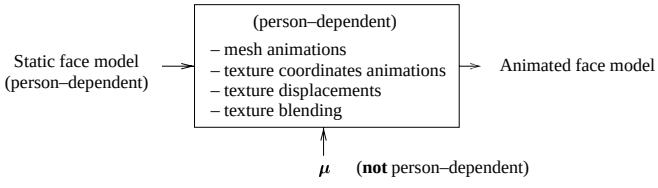

Figure 8. Synthesis part of facial animations: transforming a static model into an animated one, controlled by the general μ vector

5. Early Results: Analysis of Facial Expressions

We are presently investigating a new view–based approach to relate the analysis and the synthesis of facial expressions: using a realistic face model representing a human performer, we sample the visual space of facial expressions accross various poses, via a set of $\boldsymbol{\mu}$ vectors. Image patches for the facial features of interest (like the eyes, eyebrows, and the mouth) are extracted, to produce distinct datasets of training examples. Then, a principal component analysis is performed over those training datasets, to extract a limited number of images optimally spanning the training space. These images (called *eigenfeatures*) will allow us to characterize the facial expression of the user's face via a simple correlation mechanism, yielding a compact $\boldsymbol{\lambda}$ vector. And finally, a linear estimator will be designed to map the analysis scores $\boldsymbol{\lambda}$ to the face animation parameters $\boldsymbol{\mu}$. Such a system will not be limited by the amount of available key–frames (as noted in the introduction for traditional view–based techniques), since all degrees of freedom permitted by the synthetic face can be precisely, automatically and systematically exploited by the training strategy.

Figure 9 shows some preliminary results concerning the analysis of synthetic facial images using a linear estimator trained over the responses of the eigenfeatures. We are currently extending this strategy to the analysis of real facial images.

6. Concluding Remarks

We presented efficient and original face model construction techniques, which are useful for realistic clones.

Such realism allows to make analysis and synthesis modules cooperate to recover both the global pose and the facial expressions of a human performer, without any markers or makeup.

Our current perspectives include:

(i) the definition of an animation space sampling strategy, so that the linear estimator has enough training data to reconstruct coherent facial expressions from the analysed images, and can properly decouple the effects of the user's pose and the facial expressions from the responses of the eigenfeatures;

(ii) the extension of our analysis algorithm on real facial expressions (i.e. the mapping $\boldsymbol{\lambda} \rightarrow \boldsymbol{\mu}$ when $\boldsymbol{\lambda}$ is measured on real images, although trained on synthetic images);

(iii) and the integration of our analysis module within the global motion tracking algorithm, to have a complete face cloning framework.

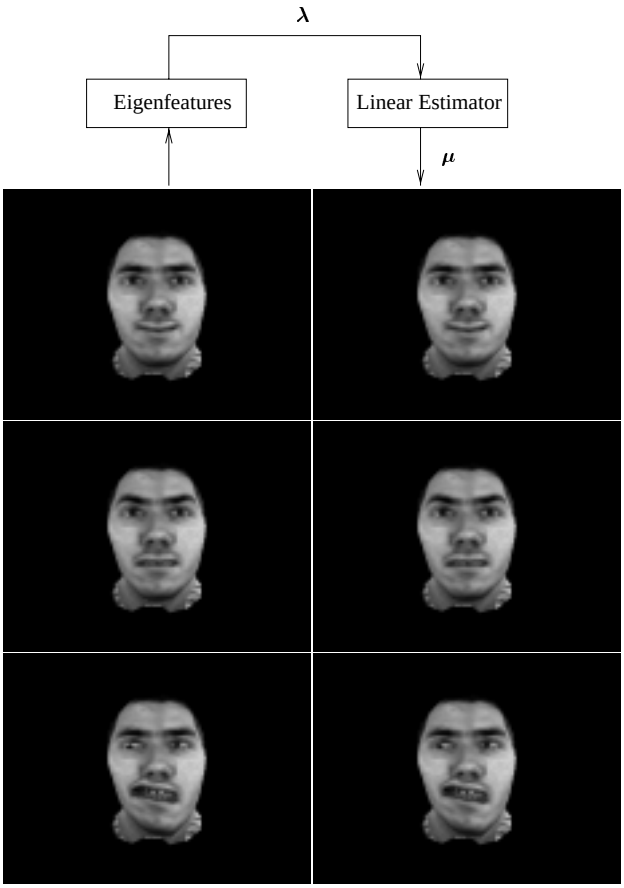

Figure 9. Some analyses of facial expressions: each image of the left column was quantified by some eigenfeatures, giving a λ vector. A linear estimator mapped λ to the animation parameters μ, which were rendered into the images of the right column

Acknowledgments

Eurécom's research is partially supported by its industrial members: Ascom, Cegetel, France Telecom, Hitachi, IBM France, Motorola, Swisscom, Texas Instruments, and Thomson CSF.

The authors would like to thank Hervé Delingette (INRIA Sophia–Antipolis) for his contribution in the model mesh construction; the University of Erlangen, Germany, and the L.U.A.P. (Université Paris–VII) for the original Cyberware scans; and Katia Fintzel (Espri Concept/Institut Eurécom) for being the model in section 4.

References

[1] B. Bascle and A. Blake. Separability of pose and expression in facial tracking and animation. In *International Conference on Computer Vision*, Bombay, India, January 4-7 1998.

[2] CYBERWARE^{TM}. URL http://www.cyberware.com

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

[4] 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.

[5] 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.

[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] 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.

[8] 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.

[9] T. Kanade, P. J. Narayanan, and P. W. Rander. Virtual*ized* reality: Concepts and early results. In *IEEE Workshop on Representation of Visual Scenes*, Cambridge, Massachusetts, June 1995. In conjonction with ICCV'95.

[10] 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.

[11] G. Maestri. Animating faces using morphs. *Digital Magic*, pages 27–28, June 1997.

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

[13] MPEG demo of the face tracking system. URL http://www.eurecom.fr/∼image/TRAIVI/valente-8points.mpg (1,782,100 bytes).

[14] 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.

[15] 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.

[16] J. Ostermann and E. Haratsch. An animation definition interface — Rapid design of MPEG–4 compliant animated faces and bodies. In *International Workshop on Synthetic–Natural Hybrid Coding and Three Dimensional Imaging*, Rhodes, Greece, September 1997.

[17] 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.

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

[19] S. Valente and J.-L. Dugelay. 3D face modeling and encoding for virtual teleconferencing. In *Workshop on Very Low Bitrate Video (VLBV'98)*, Urbana–Champaign, Illinois, October 8-9 1998. In conjunction with ICIP'98.

[20] M.-L. Viaud. *Animation Faciale avec Rides d'Expression, Vieillissement et Parole*. PhD thesis, Université de Paris XI–Orsay, Orsay, France, 1992.