

*ECCV Workshop on Perception of Human Action,*  
Freiburg, Germany, June 1998

# An Analysis/Synthesis Cooperation for Head Tracking and Video Face Cloning

Stéphane Valente<sup>1</sup>, Jean-Luc Dugelay<sup>1</sup>, and Hervé Delingette<sup>2</sup>

<sup>1</sup> Institut Eurécom  
Multimedia Communications Departement  
B.P. 193  
06904 Sophia-Antipolis Cedex, France  
{valente, dugelay}@eurecom.fr  
<http://www.eurecom.fr/~image>

<sup>2</sup> INRIA  
Projet Epidaure  
B.P. 93  
06902 Sophia-Antipolis Cedex, France  
[Herve.Delingette@sophia.inria.fr](mailto:Herve.Delingette@sophia.inria.fr)

**Abstract.** In the context of a face-cloning system, we present a head tracking algorithm based on an enhanced analysis/synthesis feedback loop which is able to handle very large rotations out of the image plane. The key idea is to make the feedback loop synthesize search patterns for the head facial features by taking into account the current face position, rotation and lighting. As pointed out by the work of A. Galowicz, analysis/synthesis cooperations are very promising, but require a high level of realism from the synthesis module. Consequently, we present geometric and photometric head modeling techniques that are realistic and computationally efficient. Finally, we reformulate a classic differential block-matching algorithm to integrate real and synthesized facial features. In addition, the feature tracking will be shown to be robust to the speaker's background, and the system performance are reported and discussed.

## 1 Introduction

After a brief presentation of face-cloning, this introduction points out the constraints that are inherent to a teleconferencing system, and how an enhanced feedback loop based on synthesized facial features can improve a face tracking algorithm.

### 1.1 Related Work

Face-cloning aims at animating a synthetic face model by analysing a video sequence of a real speaker. Basically, any cloning algorithm must address two different issues, namely the model global animation (corresponding to the speaker's

position and orientation in the 3D space), and the model local animation (showing the speaker’s current facial expressions). In the literature, many references concerning video-cloning report promising results, such as [18, 21, 16, 8, 13, 10]. Most of them assume that the speaker is looking at the camera with small head motions, and they generally use an entirely synthetic head models that represent the user in a stylish manner (*avatar*). Such “hand-made” models are popular because they are easily manipulable in real-time. A realistic physics-based model can be found in [21], where Terzopoulos and Waters consider a person-dependent Cyberware scan, and adapt it so that it can be handled by an automatic animation system. Although highly realistic and fully functional, their model is however too complex to be animated in real-time on a standard computer, and their analysis framework requires black contours to be drawn on the speaker’s face.

## 1.2 Virtual Teleconference Constraints

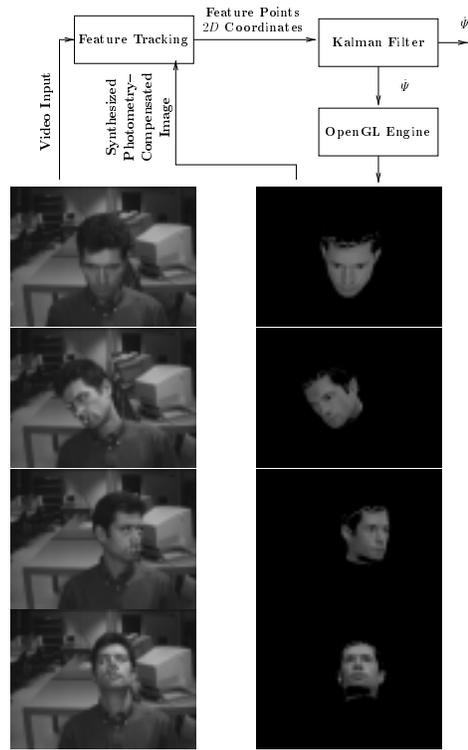
The material presented in this paper has been derived in the context of a virtual teleconferencing system [22]. Its telecommunication aspects impose specific and challenging constraints on facial cloning, like the face analysis and synthesis frame-rates, the image processing delays, the very low bandwidth networks available to transmit the animation parameters and the possibility to visualize the clone under a point of view different from the analysis camera. Moreover, if such a system has to be used outside of a laboratory, it should operate without colored marks taped on the speaker’s face, deal with unknown lighting conditions and background, allow the users to move freely in front of the camera, and yield visual results that are highly realistic. [15] provides an overview about what a virtual teleconferencing system can be useful for.

## 1.3 Principle of the Analysis/Synthesis Feedback Loop

To provide a high level of realism, we propose to use 3D texture-mapped models to represent each speaker within the virtual area. Taking advantage of this realism, we wrote a global motion tracking software implementing a modified differential block-matching algorithm tracking 2D feature points from synthesized patterns. The head tracking loop proceeds as follows (see figure 1):

- a Kalman filter predicts the head 3D position and orientation estimates at time  $t$  given the previous 2D feature points observations in all images until time  $t - 1$ ;
- using the estimated 3D parameters and the speaker’s head model, search patterns for the facial features are synthesized, hence taking into account the scale and geometric deformations that can be expected given the user’s position, and the background interference with the patterns. In addition, due to the 3D photometric compensation module described in section 2, the search patterns also reflect the expected face lighting;

- a reformulated block-matching algorithm finds the synthesized patterns in the image taken at time  $t$ ;
- the Kalman filter is then fed with the  $2D$  observations of the facial features in the image plane to produce new estimates for the head  $3D$  position and orientation at time  $t + 1$ .



**Fig. 1.** Feedback loop strategy based on a Kalman Filter and Synthetic Images —  $\hat{\Psi}$  and  $\tilde{\Psi}$  are the speaker’s  $3D$  position and orientation predicted and filtered estimates. The shown examples were extracted from a 30 seconds video sequence captured in a  $320 \times 242$  resolution at 10 frames per second.

Kalman filters are generally used in head tracking systems for two different purposes: the first one is to temporally smooth out the estimated head global parameters, as in [18], the second one is to convert the  $2D$  facial features positions observations into  $3D$  estimates and predictions of the head position and orientation [1]. In our application, the Kalman filter has one more goal: it makes the synthesized model have the same scale, position and orientation than the speaker’s face in the real view despite the acquisition by an uncalibrated camera. This is achieved by deriving the measurement equations of the filter, not

from default camera parameters, but from the perspective projection of the synthetic facial features performed by the synthesis module. When the Kalman filter is fed with the speaker’s observed facial features, it does not recover the 3D position and orientation of the user in the real world, but the 3D position and orientation of the clone in the synthetic world that will match the observations made in the real image<sup>1</sup>.

Our enhanced analysis/synthesis cooperation makes the face tracking more robust without requiring artificial marks, and supports very large rotations out of the image plane, as it can be seen on figure 1, while meeting the requirements detailed above. This paper focuses on the key points of the analysis/synthesis chain that contribute to its efficiency. In sections 2 and 3, we present geometric and photometric modeling techniques that create a face model realistic enough to be integrated in an analysis/synthesis feedback loop. Section 4 describes the influence of synthetic patterns on the block–matching formulation, including the robustness to the speaker’s background. Finally the performance of the tracking system is discussed in section 5.

## 2 Face Model Construction

In the literature, it seems that the easiest way to build a new 3D face model for a person is to start from an existing model, and to adapt it to conform the user’s face, with more or less automated algorithms, and starting from various kinds of input data. For example, one may choose to work with 2D images of a new person: Chaut *et al.* adapt by hand a spline–based generic mask using a face and profile view of the person, and texture it with pixels extracted from both views [4]. This process can be automated by image processing techniques, as in the chain described by Tang and Huang, based on the extraction of characteristic facial points [20]. In this category, we also find the work of Reinders *et al.*, who use only one view for “head and shoulders” video–coding applications in [17]. It is clear that 2D images lack information about the user’s face geometry, and as a result, such adapted models may have a poor geometric resolution.

Another approach consists in using texture and range data, obtained from cylindrical geometry Cyberware range finders [5]. Such a dataset is a highly realistic representation of the speaker’s face, but it cannot be used directly as a face model for several reasons. First, this dataset is too dense (in average 1.4 million vertices) for real–time computation. Furthermore, due to the limitation of the acquisition technology, the dataset is often incomplete and sometimes includes some outliers (as in figure 2(a)). Building a higher level face model from this kind of dataset traditionally required considerable user input, until Lee, Terzopoulos and Waters developed a framework to adapt their generic “skin and muscle” facial model to the range and texture data [12]. Although very authentic and fully functional, their model is computationally complex, and cannot be animated at interactive rates on standard workstations.

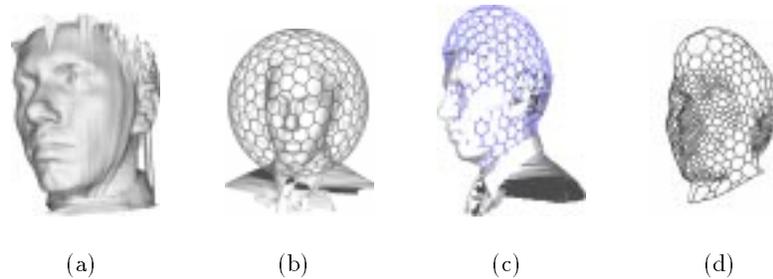
<sup>1</sup> The dynamic evolution of the filter state is trivially based on Newtonian physics with a constant acceleration assumption.

We propose an alternative face modeling technique from texture and range data, yielding models that are simpler to manipulate and animate.

## 2.1 Mesh Recovery from Range Data

To achieve both visual realism and real-time computation, we need a geometric model with a limited number of vertices but with enough details in order to distinguish facial features such as the lips or eyebrows. We have developed a reconstruction system based on deformable simplex meshes [6] to build such models from a Cyberware dataset. Unlike classic approaches, those deformable models are handled as discrete meshes, not relying on any parameterization. Because they are topological dual of triangulations, they can be easily converted as a set of triangles for display purposes or standard 3D file formats like VRML [24]. Finally, they can represent geometric models independently of their topology and they lead to fast computations.

In figure 2, we show the different stages of reconstruction from a Cyberware dataset where the hair information is missing and with some outliers. The deformable model is initialized as a sphere (figure 2(b)) and then deformed to roughly approximate the face geometry (figure 2(c)). The last stage consists in refining the mesh model based on the distance between the data and surface curvature (figure 2(d)).



**Fig. 2.** Reconstruction of a geometric model from a Cyberware dataset: (a) range data (b) initialization; (c) main deformation; (d) mesh refinement — We have interactively selected the area of interest (chin, ears, nose, lips) where the refinement is performed. The resulting mesh has 2084 vertices and was built in less than 5mns on a DEC Alphastation 233Mhz.

The face model is then texture-mapped by associating to each vertex of the simplex mesh the  $(u, v)$  texture coordinates of its closest point in the range data. Where no range data is available (at the hair level for instance), we project the vertex on the image plane through the cylindrical transformation of the Cyberware acquisition. This algorithm therefore produces an accurate geometric and texture face model.

As a conclusion, let us point out that our construction system can also initialize the deformable model to an existing face model instead of a sphere, in order to speed up the process and create different models for the same speaker in several facial expressions, while keeping the same number of primitives and the correspondances between the vertices of all models: this technique is called *mesh registration* [23].

### 3 Photometric Modeling

The goal of photometric modeling is to reduce the photometric discrepancies between the speaker’s face in the real world environment and his synthetic model directly at the 3D level, and can be seen as an alternative and elegant technique to other 2D view-based techniques, such as histogram fitting [11]. In [7], Eisert *et al.* propose an algorithm to recover the 3D position and intensity of a single infinite light source from a static view assuming an initial guess of the position prior to the motion estimation. Bozdađı *and al.* [3] have a more complex approach that determines the mean illumination direction and surface albedo to be included in their Optical Flow equation for motion estimation. Both approaches are based on a Lambertian illumination model (i.e. composed of ambient and diffuse lighting) without specular reflections and cast shadows. However, in the real world, cast shadows, and specular highlights (if the user does not have make-up), are likely to occur on a face, and will be difficult to compensate using only a single light as in the previous algorithms.

In [2], Belhumeur derives that the set of images of a convex Lambertian object under all possible lighting conditions is a cone, which can be constructed from three properly chosen images, and empirically shows that cast shadows and specular reflections generally do not damage the conic aspect of the set.

Motivated by the reconstruction possibility of an arbitrary illuminated view from several object images, we propose to recover the face illumination from a single speaker’s view by using a set of light sources at different infinite positions. The main advantage of our algorithm is that it can rely on the OpenGL industry-standard library to use hardware acceleration and compensate unknown light sources with ambient, diffuse and specular components at the 3D level in real-time. A similar idea, applied to interior design, is found in [19], where the scene global lighting is computed from the illumination of some objects painted by hand by the scene designer. In our algorithm, the synthetic scene lighting is adjusted by observing the illumination of the facial features in the real environment.

#### 3.1 Proposed Algorithm

Using OpenGL, we implemented the following general lighting equation, including ambient, diffuse and specular reflections induced by  $N$  independent infinite light sources for a 3D textured primitive, with an additional degree of freedom (a

luminance offset  $L_{\text{offset}}$ )

$$L_{\text{object}} = L_{\text{offset}} + L_{\text{texture}} \times (A_{\text{ambient}} + \sum_{i=0}^{N-1} [(\max\{\mathbf{l}_i \cdot \mathbf{n}, 0\}) \times D_i + (\max\{\mathbf{s}_i \cdot \mathbf{n}, 0\})^{\text{shininess}} \times S_i]) \quad (1)$$

where  $L_{\text{object}}$  denotes the final pixel luminance,  $L_{\text{texture}}$  the corresponding texture luminance,  $A_{\text{ambient}}$  the global ambient light intensity,  $D_i$  and  $S_i$  the diffuse and specular intensity for the  $i^{\text{th}}$  light,  $\mathbf{n}$  and  $\mathbf{l}_i$  the object normal and the  $i^{\text{th}}$  light source direction,  $\mathbf{s}_i$  the normalized bisector between the  $i^{\text{th}}$  light source direction and the viewing direction, and finally “shininess” the specular exponent controlling the size and brightness of specular highlights.

One can readily verify that the rendered image pixels values in equation 1 are linear with respect to the components of the light sources. Therefore, all the unknowns (the light source intensities, and the luminance offset if needed) can be estimated by a simple least mean square inversion for all the face pixels. The estimation process does not need to be constrained to output positive intensities, since OpenGL can deal with negative light intensities. Therefore, our algorithm consists in the following steps:

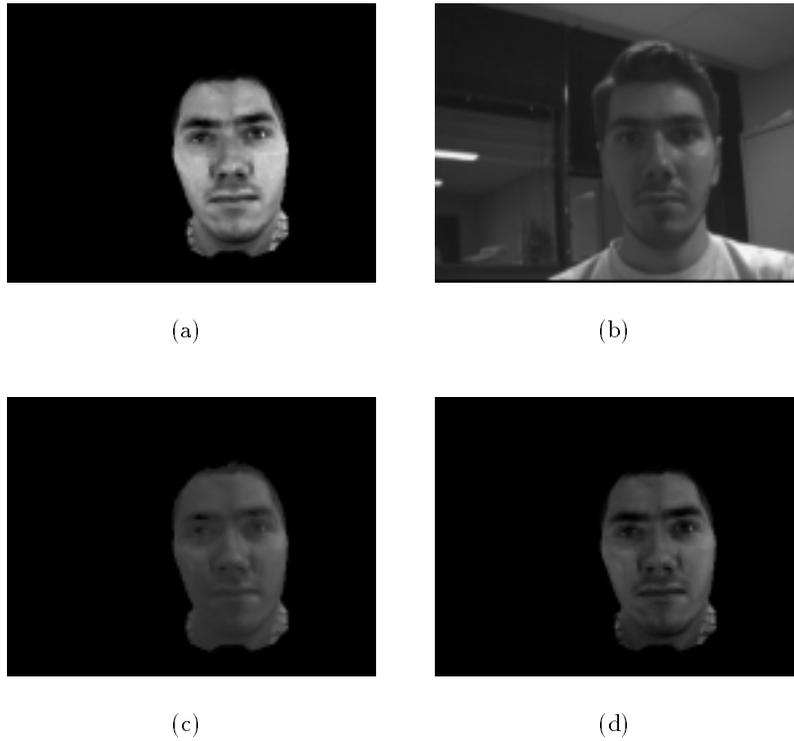
- align the synthetic model with the speaker’s image;
- extract, from the real speaker’s image, pixel luminance values around the facial features of interest. Pixels being too bright are discarded to avoid areas where the camera sensor might have saturated (the luminance of such pixels would not depend linearly on the light sources contributions);
- extract, from the synthetic image, the corresponding texture luminance values and object lighting normals;
- the light sources intensities (and the global luminance offset, if allowed) are finally estimated by solving equation 1 in the least mean square sense.

### 3.2 Results

The results of the compensation algorithm can be seen in figure 3, with and without an illumination offset. It is clear that such a compensation does not exactly match the real scene illumination, but it helps gaining consistency between the synthetic and real facial features, and allows the feature-tracking algorithm to match them correctly. Interested readers are invited to download [23] for an experimental study about the compensation performance.

## 4 Block-Matching Synthetic Facial Features

We have seen in the former sections that resorting to synthetic facial features predicted by a Kalman filter solves the problems of local geometric distortions, variations of scale and changes of lighting due to the speaker’s 3D motions. And considering that the synthetic face is rendered on a black background, it is



**Fig. 3.** Illumination compensation on a real face — from left to right: the speaker’s head model with no directional light source, the speaker in a real environment, and the same model with illumination compensation (with and without an illumination offset).

also possible to extract cues on the way facial features might overlap with other background objects during large head rotations.

However, a classic block-matching algorithm may have problems to find the correct match for two reasons: the first one is that some photometric differences might still occur between the synthesized and real facial features, and the second one is that the synthetic background might be matched against the real one. To overcome these limitations, we reformulated the block-matching algorithm.

#### 4.1 Block-Matching Reformulation

Extending the theory presented in [9], we propose to adapt the classic differential block-matching formulation (sometimes called *pattern correlation* in the literature) to handle photometric model failures using a luminance scaling and offset on synthesized features.

The differential block-matching algorithm is derived by considering that a reference pattern at time 0 denoted  $\mathbf{I}(0, 0)$  (all pixels are placed in a column vector) can undergo some perturbations  $\boldsymbol{\mu} = (\mu_1, \dots, \mu_n)^T$  (most often displacements over the image). Writing a Taylor series expansion for small perturbations between two consecutive frames, we have

$$\mathbf{I}(\boldsymbol{\mu}, \tau) = \mathbf{I}(0, 0) + \mathbf{M}\boldsymbol{\mu} + \mathbf{I}_t\tau + \text{higher-order terms} \quad (2)$$

with  $\mathbf{M} = [\frac{\partial \mathbf{I}}{\partial \mu_1}(0, 0) | \dots | \frac{\partial \mathbf{I}}{\partial \mu_n}(0, 0)]$  and  $\mathbf{I}_t = \frac{\partial \mathbf{I}}{\partial t}(0, 0)$ . Solving for  $\boldsymbol{\mu}$  in the least mean square fashion yields

$$\boldsymbol{\mu} = -(\mathbf{M}^T \mathbf{M})^{-1} \mathbf{M}^T \mathbf{I}_t \quad (3)$$

In equation 2, the  $\boldsymbol{\mu}$  perturbations are general enough to represent a local pattern rotation or scaling, and we add a luminance scaling and offset perturbation in case of a photometric model failure for a synthesized feature ( $\frac{\partial \mathbf{I}}{\partial \text{lum. scale}}(0, 0) = \mathbf{I}(0, 0)$  and  $\frac{\partial \mathbf{I}}{\partial \text{lum. offset}}(0, 0) = (1, \dots, 1)^T$ ).

This formulation is computationally efficient, because only the translation parameters<sup>2</sup> of  $\boldsymbol{\mu}$  have to be computed from (3), although other degrees of freedom can be introduced in  $\mathbf{M}$  (like luminance variations, local rotations...).

#### 4.2 Background Awareness

In figure 4, the synthesized model is rendered on a black background, which appears in the extracted rectangular nose pattern. With no special care, a classic differential block-matching algorithm is likely to match these black pixels with the image darkest areas. With a small computational overhead, our algorithm takes them into account to match only the potential feature areas in the real speaker's image.

<sup>2</sup> The Kalman Filter is fed with the 2D facial features displacements in the image plane, and does not rely on any other local parameter of  $\boldsymbol{\mu}$  to estimate the head 3D global position and orientation.



Fig. 4. Synthesized patterns are robust to the background presence.

To make the synthesized patterns be more selective (or matched against non-rectangular areas), the pattern pixels are classified into 2 subsets,  $\mathbf{I}|_F$  and  $\mathbf{I}|_B$ , whether they belong to the face or to the background area. If equation 3 is interpreted as the simple correlation of the pixel-wise difference  $\mathbf{I}_t$  between the current image and the full feature pattern with matrix  $-(\mathbf{M}^T \mathbf{M})^{-1} \mathbf{M}^T$ , then  $\mathbf{I}_t|_F$  (the pixel-wise difference restricted to the  $\mathbf{I}|_F$  subset) is the contribution of the face pixels to the general displacement  $\boldsymbol{\mu}$ .

In practice, when a background pixel is detected in a synthetic pattern, the pixel-wise difference for this pixel in  $\mathbf{I}_t$  is set to zero. Hence, the background objects (corresponding to black parts in the synthesized patterns) have no impact on the correlation score, and the algorithm finds the correct match despite other objects in the feature neighborhood.

## 5 Discussion on the Tracking Robustness

The result of our face tracking algorithm can be seen in an Mpeg sequence available on the WWW [14]. Its speed mainly depends on the workstation graphics hardware acceleration and its video acquisition speed. On a  $O^2$  SGI workstation, the analysis frame rate using 12 facial feature areas is:

- 1 image per second, when synthesizing patterns, computing the product  $-(\mathbf{M}^T \mathbf{M})^{-1} \mathbf{M}^T$ , and updating the Kalman filter for every frame;
- 10 frames per second, when disabling synthetic pattern calculation for every frame, but still enabling the Kalman filter — in this case, large face rotations might cause the system loose the user's head;
- full frame rate, when disabling both pattern synthesis and the Kalman filter — the system just tracks the facial features in  $2D$ , without recovering the head  $3D$  position and rotation, and becomes very sensitive to rotations.

In fact, the individual facial features trackers work quite well, even during large face rotations when it becomes difficult to distinguish the facial features from the scene background (look for example at the speaker's right eye on figure 4). From our experiments, the main difficulty to obtain a robust face tracking system is the tuning of the Kalman filter, which requires to set noises for the observations and the system dynamics, as a trade-off between the filter stability and its reactivity to incoming observations. The problem in our application is

that we do not have any *a priori* knowledge about its dynamic range, unlike in a more physics-based system (a radar tracking a plane for instance).

Another question that might be raised is what happens when the user closes his eyes, smiles, or does anything that differs from the static facial expression of his model: in general, the system copes with it, because it melts enough facial features observations to allow a few of them to be wrong.

## 6 Concluding Remarks

In the previous sections, we proposed a face tracking framework which has the possibility to feed the feature-tracking procedure with synthesized facial patterns. We presented geometric and photometric modeling techniques to make the synthesized patterns be closer to their real counterparts. We also reformulated a block-matching algorithm to make it work with synthetic input data, and showed how to handle the presence of the speaker's background in extreme positions. It is important to note that such an analysis/synthesis cooperation is successful because of the realism of our modeling techniques, and the design of the Kalman filter to make it control the model synthesis.

We have already obtained early results in the modeling of facial expressions, and we are presently working on how to relate them to some analysis techniques within the feedback loop. When the cloning system is complete (i.e. capable to update the model with the user's current facial expressions, and consequently the details of the synthetic patterns feeding the trackers), we expect it to be less sensitive to the Kalman filter tuning due to the gain of tracking accuracy.

## References

1. A. Azarbayejani, T. Starner, B. Horowitz, and A. Pentland. Visually controlled graphics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):602–605, June 1993.
2. P. Belhumeur and D. Kriegman. What is the set of images of an object under all possible lighting conditions? In *IEEE Conference on Computer Vision and Pattern Recognition*, November 1996.
3. G. Bozdağı, M. Tekalp, and L. Onural. 3-D motion estimation and wireframe adaptation including photometric effects for model-based coding of facial image sequences. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 246–256, June 1994.
4. P.-E. Chaut, A. Sadeghin, A. Saulnier, and M.-L. Viaud. Création et animation de clones. In *Imagina — Méta-mondes/Metaverses*, pages 244–257, Monaco, Février 1997.
5. CYBERWARE Home Page. URL <http://www.cyberware.com>.
6. H. Delingette. General object reconstruction based on simplex meshes. Technical Report 3111, INRIA, February 1997.
7. P. Eisert and B. Girod. Model-based 3D-motion estimation with illumination compensation. In *6<sup>th</sup> International Conference on Image Processing and its Applications (IPA 97)*, pages 194–198, Dublin, Ireland, July 1997.

8. I. A. Essa and A. Pentland. A vision system for observing and extracting facial action parameters. In *International Conference on Computer Vision and Pattern Recognition*, pages 76–83, Seattle, WA, June 1994.
9. G. Hager and P. Bellhumeur. Real-time tracking of image regions with changes in geometry and illumination. In *IEEE Conference on Computer Vision and Pattern Recognition*, November 1996.
10. T. S. Huang and L. Tang. Model-based video coding — Some challenging issues. In Y. Wang, S. Panwar, S.-P. Kim, and H. L. Bertoni, editors, *Multimedia Communications and Video Coding*, pages 215–221. Plenum Press, New-York, 1996.
11. T. S. Jebara and A. Pentland. Parametrized structure from motion for 3D adaptive feedback tracking of faces. In *IEEE Conference on Computer Vision and Pattern Recognition*, November 1996.
12. Y. Lee, D. Terzopoulos, and K. Waters. Realistic modeling for facial animation. In *SIGGRAPH 95*, pages 55–62, Los Angeles, California, August 6-11 1995.
13. 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.
14. Mpeg demo of the face tracking system. URL <http://www.eurecom.fr/~valente/Clonage/valente-8points.mpg>. (1782100 bytes).
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. I. S. Pandzic, P. Kalra, and N. Magnenat Thalmann. Real time facial interaction. *Displays*, 15(3), 1995. *Butterworth — Heinemann*.
17. M.J.T. Reinders, P.L.J. van Beek, B. Sankur, and J.C.A. van der Lubbe. Facial feature localization and adaptation of a generic face model for model-based coding. *Signal Processing: Image Communication*, 7:57–74, 1995.
18. 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.
19. C. Schoeneman, J. Dorsey, B. Smits, J. Arvo, and D. Greenberg. Painting with light. In *SIGGRAPH 93*, pages 143–146, Anaheim, California, August 1-6 1993.
20. L. Tang and T. S. Huang. Automatic construction of 3D human face models based on 2D images. In *IEEE International Conference on Image Processing*, Lausanne, Switzerland, September 1996.
21. D. Terzopoulos and K. Waters. Analysis and synthesis of facial image sequences using physical and anatomical models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6), June 1993.
22. S. Valente and J.-L. Dugelay. A multi-site teleconferencing system using VR paradigms. In *Ecmast*, Milano, Italy, 1997.
23. S. Valente, J.-L. Dugelay, and H. Delingette. Geometric and photometric head modeling for facial analysis technologies. Technical Report RR-98-041, Institut Eurécom, Sophia-Antipolis, France, May 1998. URL <http://www.eurecom.fr/~image/Publis98/RR-98-041.ps.gz>.
24. VRML. URL <http://vrml.sgi.com>.