Institut EURECOM[1]
2229, route des Crêtes
B.P. 193
06904 Sophia Antipolis
FRANCE

INRIA[2]
2004, route des Lucioles
B.P. 93
06902 Sophia-Antipolis
FRANCE

Research Report N° RR-98-041

Geometric and Photometric Head Modeling for Facial Analysis Technologies

Stéphane Valente[1], Jean–Luc Dugelay[1] & Hervé Delingette[2]

May 1998

Telephone: +33 (0)4 93 00 26 27
+33 (0)4 93 00 26 41
+33 (0)4 92 38 77 64
Fax: +33 (0)4 93 00 26 27

E-mail: valente@eurecom.fr
dugelay@eurecom.fr
Herve.Delingette@sophia.inria.fr

Abstract

We present geometric and photometric head modeling techniques that are computationally efficient, and yet achieve a high level of realistic animation. First, we reconstruct a textured face model from range data obtained by a Cyberware scanner. The geometric model is selectively refined at features of interest while simultaneously extrapolating missing data, and the final head model is suitable for real–time facial animation. We then propose a photometry compensation algorithm using the OpenGL graphics library, that reduces the photometric discrepancies at the $3D$ level between a synthesized view of the model and the same view of the real person. We evaluate the performance of the proposed algorithm using computer generated images.

Due to the realism of the geometric and photometric models, enhanced analysis/synthesis cooperations are made possible in such applications as face cloning, head tracking, face recognition, video indexing or person authentification.

Contents

Abstract **i**

1 Introduction **1**

2 Face Model Construction **1**

2.1 Related Work 1

2.2 Mesh Recovery from Range Data 2

2.3 Mesh Registration from Range Data 2

3 Facial Animation Possibilities **3**

3.1 Geometric Animations 3

3.1.1 Mesh Iterative Adaptation. 4

3.1.2 Mesh Edition. 4

3.2 Texture Animation 4

4 Photometric Modeling **5**

4.1 Related Work 5

4.2 Proposed Algorithm 6

4.3 Experimental Results 6

5 Concluding Remarks **8**

References **10**

1 Introduction

This article is organized as follows: section 2 presents a face model construction algorithm from range and texture data; then, section 3 deals with facial expressions modeling; in section 4, we propose a $3D$ illumination compensation (or lighting reconstruction) algorithm to match synthesized model images with real views. And finally, we discuss the potential applications of our modules in section 5.

2 Face Model Construction

2.1 Related Work

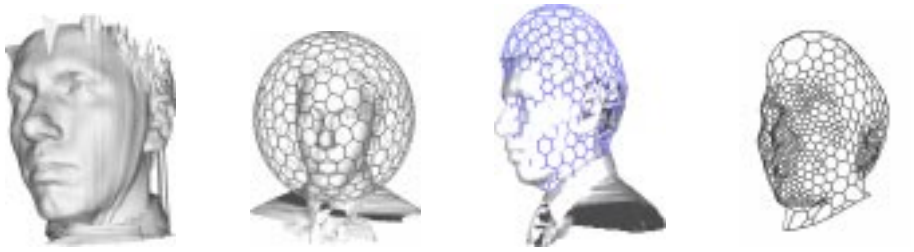
In the literature, it seems that the easiest way to build a new $3D$ face model for a person is to start from an existing model, and to adapt it to conform the user's face, with more or less automated algorithms, and starting from various kinds of input data.

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 [1]. 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 [2]. 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 [3]. It is clear that $2D$ images lack information about the user's face geometry, and as a result, such adapted models have a poor geometric resolution.

Another approach consists in using texture and range data, obtained from cylindrical geometry Cyberware range finders [5][1]. 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 1(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 [6]. Although very authentic and fully functional, their model is computationally complex, and cannot be animated at interactive rates on standard workstations.

We propose alternative face modeling techniques from texture and range data, yielding models that are simpler to manipulate and animate. We will see that our algorithms are not only able to create a new model from range and texture data with limited user interaction (section 2.2), but also able to automatically adapt an existing model to new data (section 2.3).

[1] Cyberware scanners are not the only devices capable to produce a range and texture dataset: for example, Proesmans and Van Gool developped an inexpensive system which analyses a grid projected on a face [4].



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

Figure 1: 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.

2.2 Mesh Recovery from Range Data

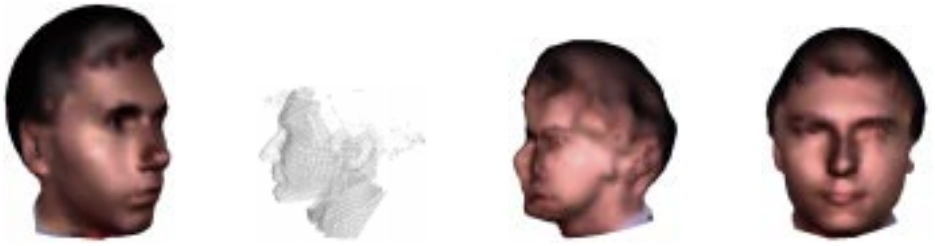
As we said in the introduction, a Cyberware dataset is not directly suitable for local deformation computation, and more generally for manipulations. To achieve both visual realism and real–time computation, we need a geometric model with a limited number of vertices but with enough details in order to distinguish facial features such as the lips or eyebrows. We have developed a reconstruction system based on deformable simplex meshes [7] to build such models. 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 [8]. Finally, they can represent geometric models independently of their topology and they lead to fast computations.

In figure 1, 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 1(b)) and then deformed to roughly approximate the face geometry (figure 1(c)). The last stage consists in refining the mesh model based on the distance between the data and surface curvature (figure 1(d)).

The face model is then texture–mapped by associating to each vertex of the simlex 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.

2.3 Mesh Registration from Range Data

In addition to recovering geometric models with a prescribed number of vertices, deformable surfaces described as simplex meshes are used to perform non-rigid reg-



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

Figure 2: Reconstruction of a geometric model for a new person starting from an existing head model representing someone else (Non-rigid Registration): (a) the initial textured simplex mesh; (b) the dataset; (c) mesh registration when only applying local deformations; (d) mesh registration with global and local deformations — (c) and (d) use the texture of model (a) mapped onto the range data (b) after registration.

istration on range data. The goal is to fit a given geometric face model on a Cyberware dataset while preserving the correspondence between them. In another words, we would like the vertices on the nose of the recovered model to be on the nose of the original model.

Our non–rigid registration method proceeds by first applying global transformations (such as rigid or affine transformations) on the reference face model to minimize the distance between the model and the dataset. Those transformations are applied iteratively as in the ICP algorithm [9]. We have then introduced a new framework (see [10]) that combines a global and a local displacement field in a simple manner. Since we proceed in a global to local manner, the method maintains the geometric correspondence between the different facial features. Figure 2(d) shows that the facial features correspondence is kept, because it uses the texture of 2(a) mapped onto the mesh model built from 2(b).

3 Facial Animation Possibilities

Let us first recall that the face geometric mesh has been refined at specific facial features which are meant to be animated locally. Our triangular patches wireframe offers two general ways to precisely generate facial expressions, via mesh vertices displacements, and via texture modification. Once again, the next discussion will be oriented by real–time and efficiency concerns.

3.1 Geometric Animations

Mesh morphing consists in interpolating the mesh vertices positions 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

(a) (b)

Figure 3: Facial expression modeling between a face model and other range data from the same speaker: (a) closed–mouthed/closed–eye face model initially aligned with the open–mouthed/open–eyed texture and range data (b) open–mouthed/open–eyed face model obtained after deformation

human face. It generally produces less unwanted effects like bulging, creasing and tearing than does facial animation created with bones, lattices, or direct manipulations [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. The two next sections will emphasize how our geometrical model can ease this task.

3.1.1 Mesh Iterative Adaptation.

Given two Cyberware images of the same speaker but with different facial expressions, we fit the same geometric model with the same number of vertices on both datasets. To get a nice synthesis of the facial expression, it is important to ensure a proper correspondence of facial features between the two models. However, the speaker's face may not be in the same position in the two range images. To compensate for displacements, we are using the non–rigid registration method described in section 2.3 (the only difference is that the source and target models represent the same speaker, with another facial expression). In figure 3, we show the registration between a face model with closed eyes and closed mouth, and a range image corresponding to open eyes and open mouth.

3.1.2 Mesh Edition.

Alternatively to the mesh iterative adaptation, if Cyberware scans are not available for all facial expressions, it is straightforward to export the wireframe in a standard $3D$ file format to modify it with commercial $3D$ modeling software [12], or to insert additional separate primitives to represent the teeth and tongue.

3.2 Texture Animation

The texture mapped onto the mesh vertices can be altered at rendition time either by switching texture patterns, or blending images. Figure 4 shows results of the model's gaze algorithm by switching between several predefined textures.

Figure 4: Gaze control by texture switching — the middle texture is the original data from the scanner, the other ones were altered to modify the gaze.

Besides switching, OpenGL is capable to blend different textures together to produce a new one. This possibility is highly interesting to fade wrinkles into the model texture at low–cost in terms of computations, compared to hard–coding them in heavy spline–based meshes [13].

4 Photometric Modeling

4.1 Related Work

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 [14]. In [15], 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. Bozdağı *and al.* [16] 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 [17], 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 [18], 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.

4.2 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_{offset})

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

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

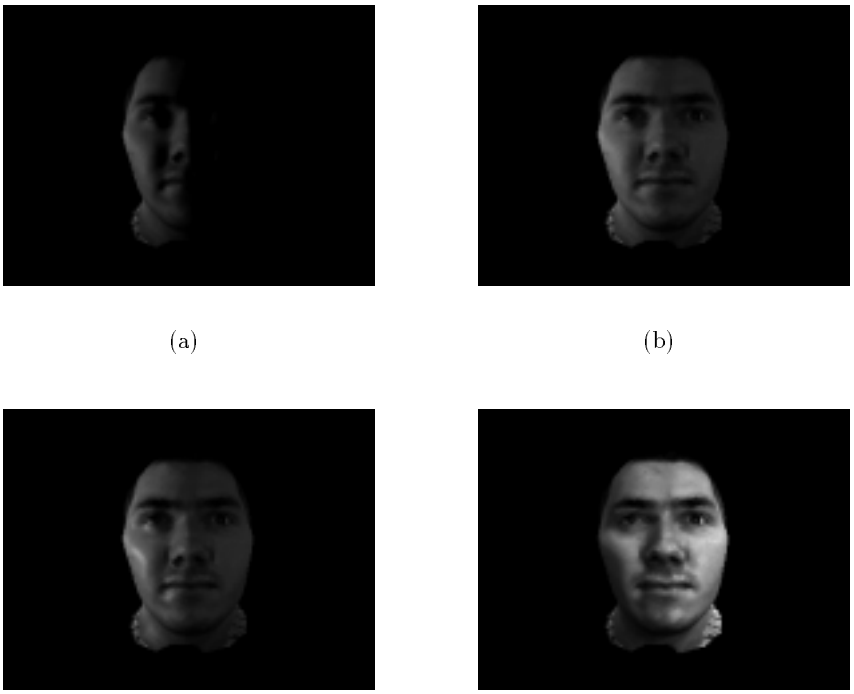
One can readily verify that the rendered image pixels values in equation 1 are linear with respect to the components of the light sources. Therefore, all the unknowns (the light source intensities, and the luminance offset if needed) can be estimated by a simple least mean square inversion for all the 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.

4.3 Experimental Results

To validate the assumption that unknown light sources can be compensated by a set of lights at predefined positions, we conducted experiments on synthetic images in order to avoid problems of misalignment between a face model and an unknown image. Four images were created respectively with a left diffuse illumination (5(a)), an ambient and left diffuse lighting (5(b)), ambient, left diffuse and specular components (5(c)), and ambient, diffuse and specular illuminations from two different light sources (5(d)).

With these images, we performed three kinds of experiments, A, B and C. In A, we compensated the face illumination with lights located at the same positions



(a) (b)

(c) (d)

Figure 5: The synthetic test images used in the validation experiments.

than the sources used to synthesize the images, enabling and disabling the luminance degree of freedom of equation 1. Such experiments can point out numerical differences between the lighting model of equation 1, implemented by floating point computations in our software, and the OpenGL lighting operations, implemented by dedicated hardware. Then, in experiments B, we tried to evaluate the quality of the compensation using light sources at predefined positions to reproduce the lighting coming from unknown directions: we compensated the face illumination with all the light sources on but the ones used for the image creation (our software has seven predefined lights, namely top, bottom, left, right, and three lights around the camera). Table 1 presents the mean error and the root of the mean square error around the model facial features. Finally, in experiment C, we show the typical compensation error for a real world case 6(b).

Experiments A suggest that roundoff errors are marginal in the algorithm, and that the equation 1 is correctly implemented by OpenGL, whereas experiments B prove that it is fairly reasonable to expect to compensate ambient, diffuse and specular reflections of unknown intensities and unknown directions by a limited set of lights at predefined locations. And at last, although the compensation square error for the real user's view is not as good as for synthetic images, our algorithm has a good performance on real faces, especially when allowing a luminance offset. We believe that it is due to the misalignment between the face and the synthetic model, which cannot be perfectly matched "by hand", to the uncalibrated acquisition camera, which does not realize a perfect perspective projection, and to the texture map

Table 1: Illumination compensation mean errors and roots of mean quadratic errors for experiments A (between synthetic views using the same lighting directions), B (between synthetic views using different lighting directions) and C (between a synthetic view and a real view).

Image	5(a)	5(b)	5(c)	5(d)	6(b)
No compensation	74.25/77.66	79.92/87.31	78.84/86.79	36.21/44.27	51.40/62.42
Exp. A	$-0.01/0.11$	$-0.01/0.09$	$-0.01/0.09$	$-0.04/0.36$	
Exp. A + offset	0.06/0.24	$-0.02/0.15$	$-0.02/0.15$	0.00/0.36	
Exp. B	0.15/0.39	0.04/4.12	$-0.02/3.56$	$-0.02/3.55$	
Exp. B + offset	0.02/0.30	$-0.20/4.12$	$-0.43/3.58$	$-0.14/3.56$	
Exp. C					$-3.98/15.12$
Exp. C + offset					$-0.02/9.31$

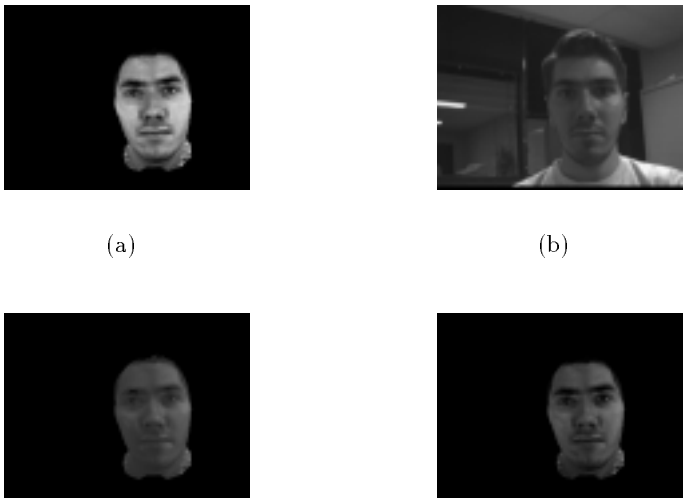
of the face model: it should actually correspond to the user's face viewed in ambient lighting, but the scanning device has built–in light sources that are powerful enough to create parasitic diffuse and specular reflections on the user's face during the texture acquisition.

We do not claim that our algorithm recovers the exact scene illumination, but it contributes to the face model realism compared to the real face view.

5 Concluding Remarks

In this paper, we proposed *geometric and photometric* algorithms to build an accurate and efficient head model of a person. These modeling techniques have been successfully used in the context of a face cloning system [19], where the speaker's face is robustly tracked by an analysis/synthesis feedback loop, without any mark or makeup on his face to highlight features of interest. The key idea of the feedback loop is to synthesize search patterns for the speaker's facial features that take into account the face $3D$ lighting and the variations of the patterns due to scale changes and large rotations out of the image plane [20]. Nevertheless, an original use of our modules could be possible for person authentification or face indexing applications. The basic principle would be to match a real face view against face models, instead of matching a real view with images taken from a database, which may not correspond to the current lighting conditions, the person's pose, and his facial expression.

Given a database of $3D$ face models, one possible procedure could be to associate the $2D$ image of the face (that must be recognized, analyzed, classified...), with one of the model. Using photometric and geometric manipulations, the first stage would consist in a "rough" alignment between the face and the synthesized views of the models. Then, the alignment could be refined using a second step of local deformations (texture and geometry) corresponding to the face expression to obtain a "full" alignment, measured by a score.



(a) (b)

(c) (d)

Figure 6: 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).

Another alternative would be to use eigenfaces generated from $3D$ models: eigenfaces are known to loose their discriminative power when the face illumination, pose and/or expression vary too much from those of the training database. After a preliminary photometric calibration stage with a known person, the illumination compensation parameters could be reused to generate $2D$ images of the face models in the database, and therefore build "on the fly" an adaptive and scalable eigenface basis, which would take into account the pose and/or the facial expressions under the current viewing conditions.

As a conclusion, our geometric and photometric algorithms can be seen as a new way to build a compact and simple representation of a head while capturing at the $3D$ level all the pose, facial expression and photometric variability of a person. For facial image analysis and recognition technologies, instead of trying to find biometric and invariant measurements, our face model could provide measurements that are adapted to a given situation by offering a scalable search space. Of course, the above–mentionned applications should be further discussed, and compared to other approaches, already published in the literature [21].

References

[1] 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.

[2] 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.

[3] 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.

[4] M. Proesmans and L. Van Gool. One–shot 3D–shape and texture acquisition of facial data. In *Audio– and Video–based Biometric Person Authentification*, pages 411–418, Crans–Montana, Switzerland, March 1997.

[5] CYBERWARE Home Page. URL http://www.cyberware.com.

[6] 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.

[7] H. Delingette. General object reconstruction based on simplex meshes. Technical Report 3111, INRIA, February 1997. ftp://ftp.inria.fr/INRIA/tech-reports/RR/RR-3111.ps.gz.

[8] VRML. URL http://vrml.sgi.com.

[9] P. J. Besl and N. D. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, February 1992.

[10] J. Montagnat and H. Delingette. A Hybrid Framework for Surface Registration and Deformable Models. In *Computer Vision and Pattern Recognition, CVPR'97*, pages 1041–1046, San Juan, Puerto Rico, June 1997.

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

[12] 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.

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

[14] 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.

[15] P. Eisert and B. Girod. Model–based 3D–motion estimation with illumination compensation. In *6^{th} International Conference on Image Processing and its Applications (IPA 97)*, pages 194–198, Dublin, Ireland, July 1997.

[16] 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.

[17] P. Belhumeur and D. Kreigman. 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.

[18] 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.

[19] S. Valente and J.-L. Dugelay. A multi–site teleconferencing system using VR paradigms. In *Ecmast*, Milano, Italy, 1997.

[20] Mpeg demo of the face tracking system. URL http://www.eurecom.fr/∼valente/Clonage/valente-8points.mpg. (1782100 bytes).

[21] Theme section. Face and gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7), July 1997.