Learning 3D Appearance Models from Video

Abstract

In the last few years, there has been a great interest in face modeling for analysis (e.g. facial expression recognition) and synthesis (e.g. virtual avatars). In this paper we introduce a semi-automatic method for 3D facial appearance modeling from video sequences, and four main novelties are proposed:

- *We introduce a 3D generative facial appearance model which takes into account the structure and appearance.*
- *Learning the appearance model in a semi-unsupervised manner from video sequences.*
- *In the learning stage, we use a flow based constrained stochastic sampling technique to improve specificity in the parameter estimation process.*
- *In the appearance learning step, we automatically select the most representative images from the sequence. This avoids biasing the linear model, speeds up the process and makes it more computationally tractable.*

Preliminary experiments of learning 3D facial appearance models from video are reported.

1 Introduction

In the last few years there has been a great interest in modeling faces for analysis (e.g. facial expression recognition) and synthesis (e.g talking heads). Among all approaches to model 3D faces from video, two of the most popular and commonly used are based on Appearance Models (AM) [12, 10, 2, 19, 5] or Rigid/Non-rigid Structure from motion (SFM) [9, 6, 18, 23]. Despite being extensively studied, both approaches suffer from several drawbacks. All SFM approaches have an implicit data conservation assumption in their formulation, since the correspondence problem is usually solved with classical trackers or optical flow techniques. In the face domain, this aspect will is dramatic as the face undergoes deep changes in appearance due to variations in expression (e.g. blinking, appearance of the tongue, etc), biasing seriously the estimation. On the other hand, AM techniques can overcome the problem of appearance changes by explicitly introduce linear variation of intensity/shape. However, AM approaches do not decouple correctly the 3D structure from the rigid/non-rigid motion due to changes in expression (all modeled with shape basis). Moreover, in order

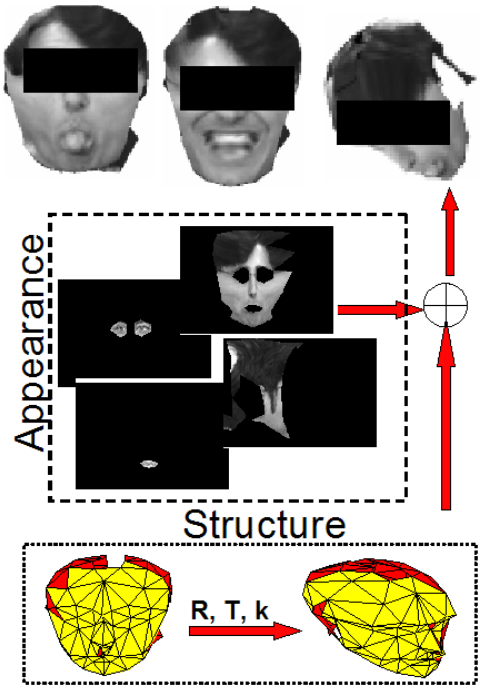

Figure 1: Generative 3D facial appearance model with structure and appearance.

to learn the appearance model, a labeled training set is usually necessary (which involves a tedious and error prone hand labeling process). In figure 1 we show some pictures illustrating the main idea of the paper.

2 Previous Work

A lot of work has been done in the area of modeling 3D faces in the past few years. It is beyond the scope of this paper to review all of them, notwithstanding we will cite the most relevant ones.

Several papers have been published recovering the 3D head motion, assuming a simple 3D model and flow equations. Basu et. al [3] use an ellipsoidal model to track the head using regularized flow. De carlo and Metaxas [13] use a generic 3D model and adjust it making use of the flow equations. Xiao et al. [21] use a cylindrical head model to recover the full 3D motion under perspective projection. Without assuming a specific 3D model, several authors have reported interesting work in the area of structure from motion (SFM). Torresani et al. [20] recover the rigid and non-rigid motion from video streams tracking individual feature points. Under orthographic projection, they are able to factorize the feature points matrix into rigid and non-rigid motion. Chowdhury and Chellappa [9] construct a 3D model by inferring depth from flow. In a similar approach but from correspondences and performing bundle adjustment, Zhang et al. [23] construct 3D models from a video sequence with the face rotation from profile to profile. Pighin et al. [18] model and animate 3D Face Models doing SFM in multi-view images and solving the correspondence by hand. Brand [6] reports a SFM technique in a new algebraic approach which allows accommodation for uncertainty and it is less prone to propagate errors.

Since Active Shape Model/Active Apperance models [10] and Morphable models [15] appeared, there has been quite a few amount of face related work in the appearance/face domain. Vetter and Blanz [5] have introduced morphable models learned from a Cyberscan which takes into account shape and texture. Romdhani and Vetter [19] have recently improved the fitting process. However, in previous work the model should be learned by a time consuming and error prone manual process. Several efficient algorithms exist to fit MM/AAM in real time [1]. Cascia et al. [8] show a method which is able to track 3D heads under changeable illumination conditions by registering w.r.t the eigenspace. De la Torre and Black [11] proposed an energy function based algorithm to learn the appearance model in an unsupervised manner. In a similar but independent work, Baker et al. [2] have proposed a method to learn the AAM in a unsupervised fashion.

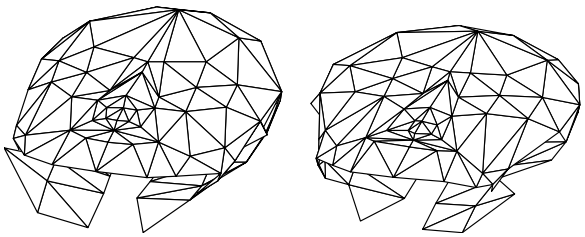

Figure 2: The original and the deformed 3D mesh.

The method we present in this paper unifies previous AM and SFM algorithms by learning the appearance model in an unsupervised fashion and having a 3D model with decoupled rigid/non-rigid parameters.

3 Generative model for 3D faces

In this section we describe a possible generative 3D facial appearance model which takes into account the structure, appearance and 3D motion.

3.1 From generic 3D structure to person-specific models

We have downloaded a generic 3d head model from (http://grail.cs.washington.edu/projects/realface/) and subsampled it. In order to give a first estimation of the shape of the face, we simply select some points by hand in two orthogonal views. The mesh is deformed with a radial basis function plus an affine transformation, such that minimizes:

$$E(\mathbf{C},\mathbf{A}) = ||\mathbf{P}_{2d} - \mathbf{CD} - \mathbf{AP}_{3d}||_F$$
$$\textit{subject to } \mathbf{C}^T\mathbf{1} = \mathbf{0} \quad \mathbf{C}^T\mathbf{P}_{3d} = \mathbf{0}$$

where $\mathbf{P}_{2d} = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \\ y_1 & y_2 & \cdots & y_n \end{bmatrix}$ are the 2D image points, $\mathbf{P}_{3d} = \begin{bmatrix} X_1 & X_2 & \cdots & X_n \\ Y_1 & Y_2 & \cdots & Y_n \\ 1 & 1 & \cdots & 1 \end{bmatrix}$ are the 3D points of the mesh. $\mathbf{A} \in \mathbf{R}^{2\times 3}$ contains an affine transformation plus translation, $\mathbf{D}$ is a matrix such that each element $d_{ij} = \exp -\frac{(X_i-X_j)^2+(Y_i-Y_j)^2}{\beta}$ is the euclidian distance [18]. Once we have re-escaled the X,Y axis, we do a similar approach to re-scale the Z axis. In figure 2.a, it is possible to see the original 3D mesh and in figure 2.b we can see the person-specific model once deformed.

3.2 Modeling appearance changes

Once the structure of the face is obtained, we construct the appearance model by mapping the 3D model into cylindrical coordinates. In figure 3.a we see how to project the mesh into cylindrical coordinates, $y = Y$ and $x = arctag(\alpha X/Z)$ where α is a variable which

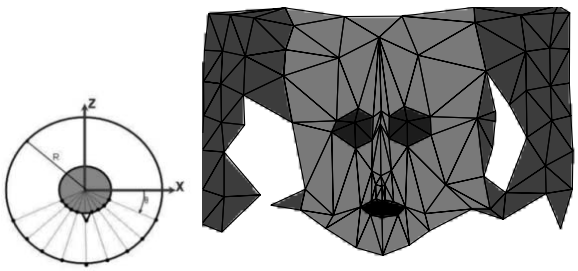

Figure 3: a)Projection into cylindrical coordinates. b)Unwarped Mesh.

Figure 4: Texture mapped from one image to the unwarped cylinder.

adjust the cylindrical projection. In figure 3.b we can see the unwarped mask.

Once we have unwarped the mesh, we map the texture from the image to the unwarped mesh, assuming perspective projection. Similar to previous work [11], in the unwarped texture image, we define four regions, corresponding to the eyes, mouth, laterals and the rest of the face. Each of the regions will contain a subspace of different dimensionality (fig. 4). Once the unwarped texture is obtained, it is mapped from the unwarped cylindrical parameter space to the 3D model, by means of the triangular patches (fig. 5).

4 Flow based initialization

We use flow based techniques to give an initial and fast estimation of the rotational and translational components of the rigid motion of the head between frames. However, flow techniques are based on the brightness constancy assumption and are well known for being noisy and ambiguous when recovering 3D information. To overcome these difficulties, we make use of robust statistics techniques [4] and approximate the average depth head with a simple parameterized 3D model. Several 3D models can be used cylindrical[21], ellipsoidal [3] or anthropomorphic models [13]. Within a

Figure 5: Two views of the texture map in 3D.

coarse-to-fine iterative strategy, we minimize[1]:

$$E(\boldsymbol{\mu}) = \sum_{p \in \mathcal{R}} \rho\big(d_{pt}(\mathbf{f}(\mathbf{x}_p, \boldsymbol{\mu})) - d_{p(t-1)}(\mathbf{f}(\mathbf{x}_p, \mathbf{0})), \sigma\big) \quad (1)$$

where $\rho(x, \sigma) = \frac{x^2}{x^2+\sigma^2}$, $\boldsymbol{\mu} = (\boldsymbol{\theta}, \mathbf{t}) = (\theta_x, \theta_y, \theta_z, t_x, t_y, t_z)$ are the parameters for the rotational and translation components and $\mathcal{R}$ is the region of support. $\mathbf{f}(\mathbf{x}, \boldsymbol{\mu})$ is the geometric transformation:

$$\mathbf{f}(\mathbf{x}, \boldsymbol{\mu}) = \begin{bmatrix} f_x \frac{\mathbf{R}(\theta_x,\theta_y,\theta_z)(X\ Y\ Z)+t_x}{\mathbf{R}(\theta_x,\theta_y,\theta_z)(X\ Y\ Z)+t_z} - x_o \\ f_y \frac{\mathbf{R}(\theta_x,\theta_y,\theta_z)(X\ Y\ Z)+t_y}{\mathbf{R}(\theta_x,\theta_y,\theta_z)(X\ Y\ Z)+t_z} - y_o \end{bmatrix} \quad (2)$$

where $\mathbf{R}(\theta_x, \theta_y, \theta_z)$ is a rotation matrix and $X\ Y\ Z$ are the 3D coordinates[2].

Minimizing expression (1) becomes a non-linear estimation problem due to the robust function and the behavior of the motion parameters. To make the problem linear we use the Iteratively Reweighted Least Squares IRLS) algorithm [16] to aproximate the optimization problem by one of weighted least squares to, at the end, linearize the estimation of the motion parameters [4]. Given an initial estimation of the motion parameters $\boldsymbol{\mu}^0$, a Gauss-Newton method can be applied by

[1]Throughout this paper, we will use the following notation: bold capital letters denote a matrix $\mathbf{D}$, bold lower-case letters a column vector $\mathbf{d}$. $\mathbf{d}_j$ represents the j-th column of the matrix $\mathbf{D}$. d_{ij} denotes the scalar in the row i and column j of the matrix $\mathbf{D}$ and the scalar i-th element of a column vector $\mathbf{d}_j$. All non-bold letters will represent scalar variables. $||\mathbf{d}||^2_{\mathbf{W}} = \mathbf{d}^T\mathbf{W}\mathbf{d}$ is a weighted norm of a vector $\mathbf{d}$. *diag* is an operator which transforms a vector to a diagonal matrix. $\mathbf{D}_1 \circ \mathbf{D}_2$ denotes the Hadamard (point wise) product between two matrices/vectors of equal dimensions.

[2]We assume that the instrinsic parameters of the camera (f_x, f_y, x_o, y_o) are known. For more details about camera calibration check (*http* : *//www.vision.caltech.edu/bouguetj/calib_doc/*).

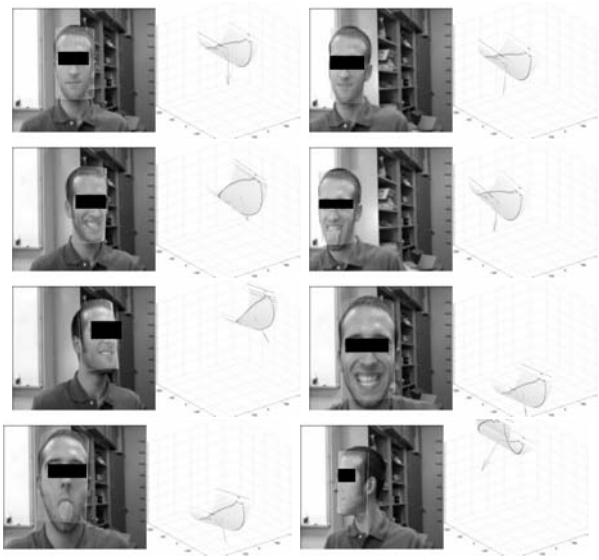

Figure 6: Some frames tracking the face under pose and facial expression changes.

incrementally updating the parameters solving the following approximate minimization problem:

$$E(\boldsymbol{\mu}) \approx ||\mathbf{d}_t(\mathbf{f}(\mathbf{x}, \boldsymbol{\mu}^0)) + \mathbf{J}_t \boldsymbol{\Delta\mu} - \mathbf{d}_{t-1}(\mathbf{f}(\mathbf{x}, \mathbf{0}))||_{\mathbf{W}_t} \quad (3)$$

where $\mathbf{J}_t = \frac{\partial \mathbf{d}_t}{\partial \boldsymbol{\mu}}$ is the Jacobian matrix. In appendix A, we provide the updating equations.

In figure 6 we can observe some images of the tracking process. Observe that the tracker can become a little bit biased due to the changes in pose/expression.

5 Dimensionality reduction

Dimensionality reduction is a common technique to filter and make algorithms more computationally tractable, specially when processing high dimensional data. When processing large videos (e.g. several minutes) the amount of redundant facial expression/poses becomes an issue for several reasons. Firstly, we do not necessarily have an uniform sampling of all the possible facial expressions/poses. This will bias the appearance learning algorithm towards reconstructing better the expressions with more samples. Secondly and more importantly, the amount of data would make the stochastic algorithm very computationally expensive. To avoid this phenomena, once the images are registered, we find the most representative prototypes by clustering, using the recent advances in multi-way normalized cuts [22] [3].

In figure 7 we show 56 prototypes extracted from a sequences of 800 frames. Figure 8 shows some of the samples of the same cluster. We can observe that individual prototypes capture changes expression/pose.

[3]The code can be download from http://www.cs.berkeley.edu/ stellayu/

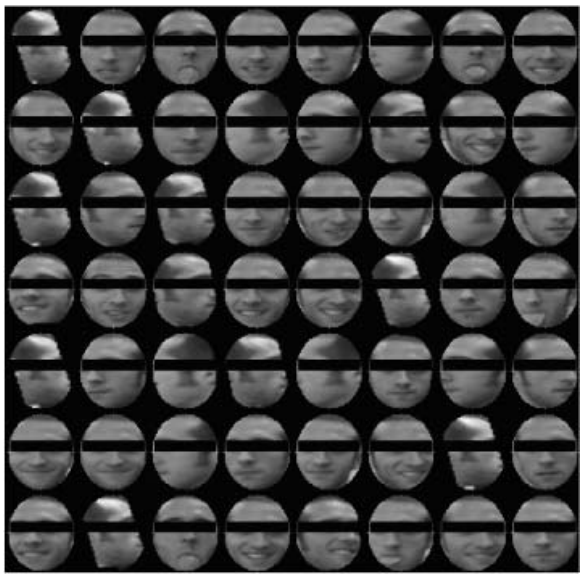

Figure 7: 56 prototypes extracted from 800 frames.

Figure 8: Samples of several clusters.

6 Stochastic Smooting for Appearance Learning

The optical flow provides us with an estimation of the rigid motion parameters, which can be biased due to changes in facial expression, the fact that the 3D model is not accurate enought and linealization error. In order to improve the estimation, compute some non-rigid motion parameters and build the appearance model, we use a smoothing particle filtering algorithm [14].

We will model an image sequence as a dynamical system and we will treat the estimation problem as one of multivariate time series analysis. In a more general sense, any dynamical system can be charac-

terized in terms of some hidden variables, the state $\mathbf{s}_t$, which summarizes the system's past behavior. The more general description of a dynamical system can be given in terms of the General State Space Model (GSSM), which can be described by the following coupled of stochastic equations:

$$\mathbf{s}_t = g(\mathbf{s}_{t-1}, \mathbf{u}_t) + \beta_t \quad (4)$$

$$\mathbf{d}_t = h(\mathbf{s}_t) + \boldsymbol{\zeta}_t \quad (5)$$

where $\mathbf{d}_t$ is the vectorized observed image frame at time t. The hidden state, $\mathbf{s}_t$, will recover $(\theta_x, \theta_y, \theta_z, t_x, t_y, t_z, \kappa)$, where κ are the non-rigid parameters (see section). $\mathbf{u}_t$ is the input to the dynamical system and β_t and $\boldsymbol{\zeta}_t$ are samples from some noise distribution. h is the *measurement function* and g describes the dynamics of the system. In the more general case they are non-linear and the noise is non-gaussian. If g and h are linear functions, and $\beta_t, \boldsymbol{\zeta}_t$ are samples from independent gaussian noise, the previous equations form the well known Kalman filter [7, 14].

6.1 Measurement Equation

Eq.(5) is the measurement equation, and expresses the fact that an image sequence at time t, $\mathbf{d}_t$, is generated by a general non-linear function h of $\mathbf{s}_t$. The likelihood of a particular sample of $\mathbf{s}_t$ is related to the image by:

$$\mathbf{M}^* = NR\big(\mathbf{R}(\theta_x, \theta_y, \theta_z) * \mathbf{M} + [t_x\ t_y\ t_z]^T, \kappa\big)$$

$$p(\mathbf{d}_t|\boldsymbol{\mu}, \kappa) \sim exp - \frac{||\mathbf{d}_t - Rec(\mathbf{d}_t(Proj(\mathbf{M}^*)))||}{\sigma} \quad (6)$$

where we define several operators; $\mathbf{M} = \begin{bmatrix} X_1 & \cdots & X_n \\ Y_1 & \cdots & Y_n \\ Z_1 & \cdots & Z_n \end{bmatrix}$ is the centered 3D mesh. $NR(\mathbf{M}^*, \kappa)$ is an operator which takes the 3D mesh and deforms the non-rigid parameters κ. κ is a vector of 3 parameters which modify the positions of eyebrows, mouth corners and the mandible aperture. Proj is the perpective projection operator $[f_x X/Z - xo\ , f_y Y/Z - yo]$ of the visible triangles in the 3D mesh. Given the projected visible triangles, Rec takes the image triangles, projects it into cylindrical coordinates and reconstruct the subspace as $\sum_{l=1}^{L} (\boldsymbol{\pi}^l \circ \mathbf{B}^l \mathbf{c}_t^l)$. Where:

- $\boldsymbol{\pi}_t^l$: Binary mask of the l layer at time t, which represents its spatial domain. $\boldsymbol{\pi}_t^l = [\pi_{1t}^l\ \pi_{2t}^l\ ...\ \pi_{dt}^l]$, where each $\pi_{pt}^l \in \{0, 1\}$ and $\sum_l \pi_{pt}^l = 1\ \ \forall p, t.$. It is defined by hand.
- $\mathbf{c}_t^l$: Coefficients which linear combination of the basis $\mathbf{B}^l$ will reconstruct the graylevel of the layer l.
- $\mathbf{B}^l$: Appearance basis of the l layer.

Observe that equation (6) represents a pseudo-likelihood (not necesarily normalized). A better measure would be achieved by probabilistic PCA [17], but due to the fact that we have much less samples than pixels, this measure can become unstable.

6.2 State Equation

Equation (4) describes the dynamical behavior of the hidden states of the dynamical system (the image sequence). In the more general case $g(\mathbf{s}_t, \mathbf{u}_t)$ is a non-linear transformation (e.g. a mixture of gaussians, a multi layer perceptron network, etc) and β_t is non-gaussian noise.

The optical flow has given a first estimation of the 3D rigid parameters, although it is well known, we have the results of the translation up to a scale factor due to the ambiguity between translation and depth. Despite the fact that the estimation of the flow can be a little bit biased, we use it to guide the search while sampling the posterior distribution of the state parameters. We combine both estimations with their covariances in a optimal Bayesian way:

$$p(\mathbf{s}_t|\mathbf{s}_{t-1}, \mathbf{f}_t) = N(\Sigma_t^{-1}(\Sigma_d^{-1}\mathbf{A}\mathbf{s}_{t-1} + \Sigma_f^{-1} * \mathbf{f}_t), \Sigma_t) \quad (7)$$

$$\Sigma_t^{-1} = \Sigma_d^{-1} + \Sigma_f^{-1}$$

where Σ_d is the uncertainty comming from the dynamical system, $\mathbf{f}_t$ is flow estimation for the rigid parameters, Σ_f is the uncertainty of the optical flow computed. To compute an estimation of Σ_f, we run several iterations of Gauss-Newton with IRLS method, and, once it has converged, we recompute the Jacobian $\mathbf{J}_t$ with the final parameter values $\mathbf{f}_t$ and a binary weighting matrix $\mathbf{W}_t$ is constructed. Then, an estimation of the uncertainty is given by $\Sigma_f = trace(\mathbf{W}_t)(\mathbf{J}_t^T \mathbf{W}_t \mathbf{J}_t)^{-1}$. $\mathbf{A}$ stands for a simple linear dynamical model, which is assumed to have a constant velocity model. Once the parameters are known, we sample from multidimensional gaussian 8 to generate new samples.

6.3 Deterministic gradient learning

Once we have a first reasonable assessment of the rigid/non-rigid parameters over a set of k frames, we unwarp the texture and compute an estimation of the subspace for each region of the face. For each unwarped frame, we have an image $\mathbf{p}_t \in \Re^{k_t \times 1}$ and a weighting matrix $\mathbf{w}_t \in \Re^{k_t \times 1}$. We minimize $E(\mathbf{B}^1, \mathbf{C}^1, \cdots, \mathbf{B}^l, \mathbf{C}^l) = ||\mathbf{W} \circ (\mathbf{P} - \sum_{l=1}^{L} \boldsymbol{\pi}^l \mathbf{B}^l \mathbf{C}^l)||_F$, where $\mathbf{W} = [\mathbf{w}_1\ \cdots\ \mathbf{w}_k] \in \Re^{k_t \times k}$ is a matrix such that $w_{ij} = 1$ is the pixel which is visible and $w_{ij} = 0$ if not. $\mathbf{B}^l \in \Re^{d \times k}$ is the set of k basis and $\mathbf{C}^l = [\mathbf{c}_1^l\ \cdots\ \mathbf{c}_n^l] \in \Re^{k \times n}$ are the set of coeficients for the l layer. We recursively update the basis to preserve

85% of the energy. Using a two step method which alternates between minimizing $\mathbf{C}$ in closed with $\mathbf{B}$ fixed and then fix $\mathbf{C}$ and optimize w.r.t $\mathbf{B}$. We do this for each of the layers. See [11] for more details.

7 Experiments

Figure 9 shows some pictures with the tracking results, after aligning the head w.r.t. the learned subspace.

showing the projected 3D mesh into the images after the smoothing and appearance learning algorithm are performed. The original sequence has approximately 800 frames from which, after tracking with flow (section 4) and clustering, 130 frames are selected. Once this reduced set of frames is selected, the stochastic algorithm is run so as to improve the Rigid Motion parameters as well as the Non-Rigid ones. From each of the 130 frames, we have taken subsets of 15 frames and run the smoothing for Condensation for registering w.r.t the subspace that we have learned previously. Good results have been gotten by using 700 particles and, tipically, 3 runs going backward and forward. The algorithm has been implemented in a non-optimized matlab code and takes roughly 7 hours for processing the original image sequence of 800 frames. This takes into account the Optical Flow, Condensation, Smoothing for Condensation and the learning of the Appearance Model.

8 Conclusions and Future Work

In this paper we have introduced a new generative model for 3D faces which takes into account structure, appearance and 3D motion. The model is learned in a semi-supervised manner with a mixture of deterministic and stochastic techniques. However, the method gets sometimes stuck in local minima and more research needs to be done to alleviate this situation. Also, more effort needs to be done in order to speed up the learning process.

We are extending this work by updating the structure of the model. Also, we are working on recognizing Facial Action Units (FACs) with this algorithm. We will learn models of action units and use them to improve the recognition performance in video sequences where the head is moving arbitrarily.

Acknowledgements.

This work has been supported by ... Withheld for review.

Appendix A

In order to compute an estimation of the 3D motion parameters, we need to compute the Jacobian $\mathbf{J}_t$ in equation 3. It is a common assumption to approximate

Figure 9: Tracking the rigid motion of the face.

the 3D rotation for its differential value, i.e.:

$$\mathbf{R}(\theta_x, \theta_y, \theta_z) \approx \begin{bmatrix} 1 & -\theta_z & \theta_y \\ \theta_z & 1 & -\theta_x \\ -\theta_y & \theta_x & 1 \end{bmatrix}$$

$$\mathbf{J}_t = \begin{bmatrix} \nabla d_{1t}^T(\mathbf{f}(\mathbf{x}_1, \boldsymbol{\mu}_t^0)) \frac{\partial \mathbf{f}(\mathbf{x}_1, \boldsymbol{\mu}_t^0)}{\partial \boldsymbol{\mu}_t} \\ \cdots \\ \nabla d_{dt}^T(\mathbf{f}(\mathbf{x}_{d_l}, \boldsymbol{\mu}_t^0)) \frac{\partial \mathbf{f}(\mathbf{x}_d, \boldsymbol{\mu}_t^0)}{\partial \boldsymbol{\mu}_t} \end{bmatrix}$$

where $\nabla d_{it}(\mathbf{f}(\mathbf{x}_i, \boldsymbol{\mu}_t^0)) = [\frac{\partial d_{it}(\mathbf{f}(\mathbf{x}_i, \boldsymbol{\mu}_t^0))}{\partial x} \quad \frac{\partial d_{it}(\mathbf{f}(\mathbf{x}_i, \boldsymbol{\mu}_t^0))}{\partial y}]^T$, is the spatial gradient of the image $\mathbf{d}_t$ warped with $\boldsymbol{\mu}_t^0$ at the position $\mathbf{x}_i$. $\frac{\partial \mathbf{f}(\mathbf{x}_i, \boldsymbol{\mu}_t^0)}{\partial \boldsymbol{\mu}_t} \in \Re^{2\times 6}$ is the derivative of the parametric motion w.r.t. the motion parameters evaluated at the pixel $\mathbf{x}_i$ and motion parameters $\boldsymbol{\mu}_t^0$.

$$\frac{\partial \mathbf{f}}{\partial \boldsymbol{\mu}_t} = \begin{bmatrix} \frac{Yh_1}{h_3^2} & \frac{Xh_3+Xh_1}{h_3^2} & \frac{Y}{h_3} & -\frac{1}{h_3} & 0 & \frac{h_1}{h_3^2} \\ \frac{Zh_3+Yh_2}{h_3^2} & \frac{Xh_2}{h_3^2} & -\frac{X}{h_3} & 0 & -\frac{1}{h_3} & \frac{h_2}{h_3^2} \end{bmatrix}$$

Figure 10: Some poses of the 3D learned mesh

where $h_1 = Z(1-\theta_z y+\theta_y)+t_x$, $h_2 = Z(-\theta_z x+y-\theta_x)+t_y$, $h_3 = Z(-\theta_x x+\theta_y y+1)+t_z$, $x = X/Z$ and $y = Y/Z$. Also, the first row of $\frac{\partial \mathbf{f}}{\partial \boldsymbol{\mu}_t}$ has to be multiplied by $-f_x$ and the second one has to be multiplied by $-f_y$. Once the linealization is computed, solving for eq. 3 only consist in solvind a linear system of equations:

$$(\mathbf{J}_t^T \mathbf{W}_t \mathbf{J}_t)\boldsymbol{\Delta\mu}_t = \mathbf{J}_t^T \mathbf{W}_t(\mathbf{d}_t(\mathbf{f}(\mathbf{x},\mathbf{0})) - \mathbf{d}_{(t-1)}(\mathbf{f}(\mathbf{x},\boldsymbol{\mu}_t^0)))$$

where the matrix $\mathbf{W}_t$ contains the weighting factors of the IRLS. Given a residual value $\mathbf{e}_i$ and a σ value, the weights are defined as $\mathbf{W}_i = diag(\frac{\psi(\mathbf{e}_i)}{\mathbf{e}_i})$, where we consider the point by point division between two vectors and $\psi(\mathbf{x}) = \frac{\partial \rho(\mathbf{x},\sigma)}{\partial x} = [\frac{\partial \rho(x_1,\sigma)}{\partial x_1}, ..., \frac{\partial f(x_d,\sigma)}{\partial x_d}]^T$.

References

[1] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 2004.

[2] S. Baker, I. Matthews, and J. Schenider. Image coding with active apperance models. Technical Report TR-03-13, CMU-RI, 2003.

[3] S. Basu, I. Essa, and A. Pentland. Motion regularization for model-based head tracking, 1996.

[4] M. J. Black and A. D. Jepson. Eigentracking: Robust matching and tracking of objects using view-based representation. *International Journal of Computer Vision*, 26(1):63–84, 1998.

[5] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In . *SIGGRAPH*, pages 187–194, 1999.

[6] M. Brand. 3d morphable models from video. In *Conference on Computer Vision and Pattern Recognition*, 2001.

[7] A. E. Bryson and Y. Ho. *Applied optimal control*. Blaisdell, 1969.

[8] M. Cascia, S. Sclaroff, and V. Athitsos. Fast, reliable head tracking under varying illumination: An approach based on registration of texture-mapped 3d models, 1999.

[9] A. K. Chowdhury and R. Chellappa. Registration of partial 3d models extracted from multiple video streams. In *IEEE International Workshop on Multimedia and Signal Processing*, 2002.

[10] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. In *European Conference Computer Vision*, pages 484–498, 1998.

[11] F. de la Torre and M. J. Black. A framework for robust subspace learning. *International Journal of Computer Vision*, 1(54):117–142, 2003.

[12] F. de la Torre and M. J. Black. Robust parameterized component analysis: theory and applications to 2d facial appearance models. *Computer Vision and Image Understanding*, 91(1/2):53–71, 2003.

[13] D. DeCarlo and D. Metaxas. The integration of optical flow and deformable models with applications to human face shape and motion estimation. In *Conference on Computer Vision and Pattern Recognition*, pages 231–238, 1996.

[14] M. Isard and A. Blake. A smoothing filter for condensation. In *European Conf. Computer Vision*, 1998.

[15] M. J. Jones and T. Poggio. Multidimensional morphable models. In *International Conference on Computer Vision*, pages 683–688, 1998.

[16] G. Li. Robust regression. In D. C. Hoaglin, F. Mosteller, and J. W. Tukey, editors, *Exploring Data, Tables, Trends and Shapes*. John Wiley & Sons, 1985.

[17] B. Moghaddam and A. Pentland. Probabilistic visual learning for object representation. *Pattern Analysis and Machine Intelligence*, 19(7):137–143, July 1997.

[18] F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, and D. H. Salesin. Synthesizing realistic facial expressions from photographs. *Computer Graphics*, 32(Annual Conference Series):75–84, 1998.

[19] S. Romdhani and T. Vetter. Efficient, robust and accurate fitting of a 3d morphable model. In *In International conference on Computer Vision*, pages 59–66, 2003.

[20] L. Torresani, D. Yang, E. Alexander, and C. Bregler. Tracking and modeling non-rigid objects with rank constraints, 2001.

[21] J. Xiao, T. Kanade, and J. F. Cohn. Robust full-motion recovery of head by dynamic templates and re-registration techniques.

[22] S. X. Yu and J. Shi. Multiclass spectral clustering. In *ICCV*, 2003.

[23] Z. Zhang, Z. Liu, D. Adler, M. Cohen, E. Hanson, and Y. Shan. Robust and rapid generation of animated faces from video images: A model-based modeling approach. Technical Report MSR-TR-01-101, Microsoft Research, October 2001.