A Probabilistic Framework for Rigid and Non-rigid Appearance Based Tracking and Recognition

Fernando De la Torre † Yaser Yacoob ‡ Larry Davis ‡
†Departament of Comunications and Signal Theory. La Salle School of Engineering.
Universitat Ramon LLull. Passeig Bonanova 8. Barcelona 08022 Spain
ftorre@salleURL.edu, http://www.salleURL.edu/~ftorre
‡Computer Vision Laboratory, University of Maryland, College Park, MD 20742. USA.
yaser@cs.umd.edu, http://www.umiacs.umd.edu/~yaser lsd@umd.edu, http://www.umiacs.umd.edu/~lsd

Abstract

This paper describes an unified probabilistic framework for appearance based tracking of rigid and non-rigid objects. A spatio-temporal dependent shape/texture Eigenspace and mixture of diagonal gaussians are learned in a Hidden Markov Model(HMM) like structure to better constrain the model and for recognition purposes. Particle filtering is used to track the object while switching between different shape/texture models. This framework allows recognition and temporal segmentation of activities. Additionally an automatic stochastic initialization is proposed, the number of states in the HMM are selected based on the Akaike Information Criterion and comparison with deterministic tracking for 2D models is discussed. Preliminary results of eye-tracking, lip-tracking and temporal segmentation of mouth events are presented.

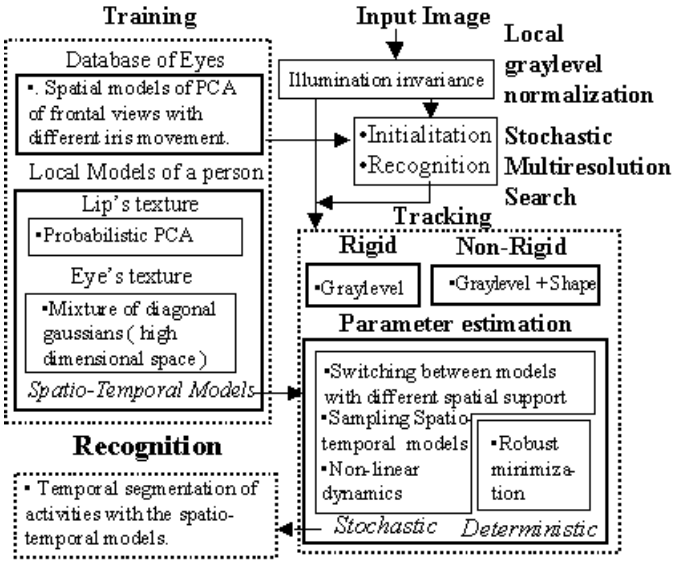

Figure 1. System Overview

1 Introduction

Recent advances in computer vision lead to new ways of interacting with computers. Fundamental tasks to be solved for vision-based human computer interaction are: detecting the presence of users and these relevant body parts, tracking faces and bodies, and analysis of gestures/expressions. In recent years several techniques have been proposed for tracking rigid and non-rigid objects. Faces are a good example where rigid and non-rigid motion is presented [1, 2, 7, 8, 16]. Black and Yacoob [2] track the non-rigid motion of the mouth, eyes and nose with local parametrized models of image motion, after the rigid motion has been eliminated. Bascle and Blake [1] track and decouple rigid and non-rigid motion of the face using active contours. La Cascia and Sclaroff [7] propose a 3D head tracker where the face is modelled as a texture map, and the tracking problem is posed as one of image registration.

All previous techniques such as correlation [2, 7], active contours[1] , and so on can fail when out of plane 3D rotations are performed or abrupt changes are produced due to appearance of the tongue or teeth. For solving such problems Black and Jepson [4] proposed Eigentracking, where all possible configurations of the object to track are registered in an eigenspace. Similar in spirit, Edwards et al. [8] construct a shape-texture eigenface called Active Appearance Models (AAM). The tracking problem consists of aligning and reconstructing the new face in the actual image with the learned texture-shape model. AAM could have a serious bias in the estimation of the parameters when 3D

changes in view are produced or with the appearance of the teeth, movement of the eyes, etc. It is not easy from a computational viewpoint to register in the eigenspace all the possible configurations of the eyes, mouth, etc, since they have independent motion[16].

Additionally we are interested in performing on-line recognition of activities within the same framework. Activity recognition using PCA has been developed by Yacoob and Black[27]. Isard and Blake[15] who proposed condensation for on-line switching between different motion models and simultaneous recognition. Recently Black [3] has extended this approach allowing spatio-temporal models of optical flow.

Figure 1 describes an overview of the proposed system. The system is composed of three modules: *Training, Tracking* and *Activity Recognition*. The training module creates a global model of eyes (GME) for detection using PCA. This module also constructs local models (LM) of the eyes and lip's texture, one for each person. In the first frame, after detecting the eye with GME, a recognition system recognize the user in front of the camera based on his eye. Once the person is recognized the LMP of this person is used for tracking. The tracking part of the system allows stochastic parameter estimation within Bayesian framework with condensation algorithm. It also allows robust deterministic tracking of rigid and non-rigid motion. The deterministic tracking achieves robust parameter estimation, while the stochastic search allows working with local model with different spatial support, automatic initialization and sampling from spatio-temporal models (STM) on line. Once the tracking is performed, recognition of the activities previously learned in the STM is possible within the HMM framework.

In sum, this paper describes a unified approach for tracking rigid and non-rigid objects using deterministic and stochastic techniques. It differs from previous approaches in that it integrates tracking with changes in pose, iconic changes and non-rigid motion of the object, within a probabilistic framework. Additionally, a temporal structure for a combined shape-texture Eigenspace is learned for constraining the parameter search-space and simultaneous recognition. It extends and unifies previous work [4] allowing flexible motion, local Eigenspaces with different spatial support, switching and temporal constraints within a stochastic framework. It also extends previous work on online recognition [3, 15], using appearance based representation and non-linear dynamics for particle filtering.

2 Learning Local Models

Since low-dimensional parametrization for representing faces have been proposed by Sirovich and Kirby [22], several authors have used this idea to parametrize subspace representations of faces, shapes, motion, etc [3, 4, 8, 25, 19, 27] based on PCA. Learning manifold representations from training data is an important part in the construction of parametric models. One drawback of PCA is that it does not capture temporal ordering in the data, even though temporal information is essential in the recognition of temporal events such as gestures or speech reading. Another drawback is that PCA assumes a multidimensional gaussian manifold [13], which imposes a restriction when data with non-smooth changes need to be learned in the same model. On the other hand, when many changes in appearance are possible it is impractical from a computational view point to work with one Eigenspace [10]. PCA also lacks likelihood model [24]. For these reasons, we developed parametric models that have the properties of incorporating temporal information and breaking non-linear manifolds into local linear regions to better constrain the model. Bregler and Omohundro [6] proposed a method for learning nonlinear surfaces from data which divide the data into overlapping clusters and performs a local PCA in each cluster. This method has been applied by Heap et al. [13] to constrain a Point Distribution Model (PDM) and extended in [12] to deal with discontinuities in shape. In this section we extend it allowing different spatial support between the samples and embedding the tracker in a temporal structure (HMM), which allows simultaneous tracking and recognition. Hidden Markov Models(HMMs) offer the possibility of modeling the observation space as a dynamically evolving mixture model, where mixing probabilities in each time-step are conditioned on those of the previous time-step via the transition matrix. This offers the possibility of non-linear spatio-temporal manifold learning.

2.1 Learning the Texture Manifold

The texture of the object to be tracked usually forms a high dimensional manifold, which makes all the density estimation techniques numerically unstable because lack of enough training data The approach explored here uses high dimensional input vectors and in order to define properly a likelihood model, we fit a mixture of gaussians with diagonal covariance matrix. Note that this constraint does not avoid the model from capturing pixel dependencies since multiple gaussians distributions are allowed. The mixture of high dimensional gaussians constrains the person's model better than PCA, since non-linealities in the manifold (produced due to changes in pose) make the distribution be likely multimodal. Once the temporal data is gathered and normalized, a HMM with the standard Baum-Welch algorithm can learn this temporal structure and cluster the training set into independent texture representations with transition probabilities between them, similar to [26]. Since the spatial support of the observation space can change due to

changes in pose (e.g 3D changes), we associate a binary mask with every realization of the appearance sample. This mask will be 1 if the texture is presented and 0 otherwise. More proper weighted of the mask could be used [7, 21], but this works well for our purposes.

In figure 2 some training images for tracking both eyes are shown. In the same figure the means of some states in the HMM are shown, also the covariance matrix is drawn. Observe how the covariance is higher in the regions with iris motion.

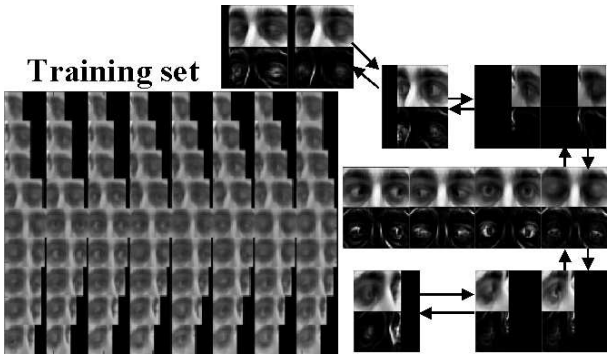

Figure 2. Some training data with temporal order. The mean and diagonal covariance values of the states in the HMM.

Each state i in the HMM has associated a gaussian or mixture with a mean $\boldsymbol{\mu}_i$, a diagonal variance matrix $\mathbf{D}_i$ and a binary mask $\mathbf{W}_i$. At this point we have constructed local models with different spatial support, represented by $(\boldsymbol{\mu}_i, \mathbf{D}_i, \mathbf{W}_i)$ with transition probabilities between them.

2.2 Learning Dependent Manifolds of Shape and Texture

In this section we describe a method for modeling a shape space with a dependent texture model. We compute a non-linear shape space as [6, 13] do, but we model the covariance in the shape-space and associate with it a dependent texture model.

PDMs [9] provide a method for representing flexible objects by means of a set of feature points describing a deformable shape. An object's shape is represented as a $2m$-dimensional vector of m image coordinates, $\mathbf{s}_i = [x_{i0}, y_{i0}, x_{i1}, y_{i1}, \cdots, x_{i(m-1)}, y_{i(m-1)}]^T$. A matrix $\mathbf{A}$, the columns of which form the training set, is factorized[1] by SVD $\mathbf{A} = \mathbf{U}^S \boldsymbol{\Sigma}^S (\mathbf{V}^S)^T$. Throughout this paper, we use $\mathbf{U}^G$ and $\mathbf{U}^S$ for denoting the orthogonal basis for greylevel, and shape modes. Every shape can be expressed approximately as the mean $\overline{\mathbf{x}}$ plus a linear combination of k columns of $\mathbf{U}^S$. We extend PDMs in the following way: each feature point in the PDM will have some graylevel or filter response value [11] associated with it. In figure 3 a possible graylevel neighborhood for each characteristic point at different multiresolution levels is presented.

[1] After subtracting the mean $\overline{\mathbf{x}}$

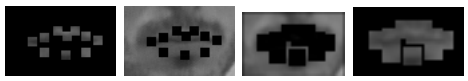

Figure 3. Some graylevel representation at different scales for each shape

To construct the coupled STM we would like each gaussian or mixture of gaussians in one state in the HMM to represent a generative model of one mouth activity (e.g speaking, smiling, etc). Thereby, tracking and recognition will be done simultaneously. Applying the standard Baum-Welch algorithm to the training data does not guarantee such a condition, since maximizing the likelihood does not necessarily fit the gaussians to the generative models. Unlike traditional unsupervised techniques which do not guarantee a satisfactory solution, we propose a simple supervised method for computing these generative models of mouth events. Given the training set, we manually label the mouth events by their actions: speaking, smiling, being surprised, null, etc. Once this labeling has been done, a dimensionality reduction is performed on the whole training set via PCA. Given the labeled data of each event, we can cluster each event in the Eigenspace with a gaussian or mixture of gaussians using the EM algorithm [18]. In fig. 4 left we fit a mixture of gaussians to four different activities, each of them represented by a different symbol. In figure 4 right the standard EM for a mixture of gaussians fails to capture these generative factors. Once this generative mixture is constructed,

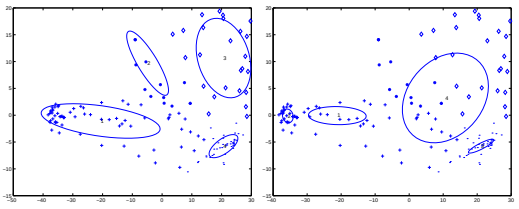

Figure 4. Generative model of mouth events fitted in supervised manner. Unsupervised clustering with 4 clusters.

we estimate the transition matrix A and prior parameters π_0 of the HMM as the traditional Baum-Welch algorithm does, but without re-estimating the mixture parameters.

Each state i in the HMM from a STM j, covers some allowable shape space represented by $(\boldsymbol{\mu}_{ji}^S, \boldsymbol{\Sigma}_{ji}^S)$, and has associated with it an independent texture representation. This

local Eigenspace is represented by a $(\boldsymbol{\mu}_{ij}^G, \mathbf{U}_{ij}^G, \eta)$. η is the average of the eigenvalues in the orthogonal subspace of the first eigenvectors [19].

2.3 Selecting the number of states

In practice, the number of states in the HMM is typically unknown and usually is assigned by hand. The most succesfull method for solving this problem is the Minimum Description Length (MDL) [23]. Recently Brand [5] has proposed an entropic model for structure discovery which could be seen as an instance of MDL when a entropic non-informative prior is used. If just the maximum likelihood is used as a measure of how well a model fits the data, it tends to over-parameterize the representation, using a prior can remove this deficiency. The Akaike Information Criterion (AIC) [23] and MDL do in spirit the same thing, they operate on the principle that once a model has an adequate number of parameters to describe the data, any increase in the number of parameters does not usually result in a very significant increase in the likelihood. Therefore a term that penalizes complexity is subtracted from the log likelihood and can result in a good compromise between number of parameters and accuracy of fit.

Here we use AIC as a way to automatically select the number of states in HMM. The AIC in a HMM is:

$$-2\log \sum_{i=1}^{States} \alpha_T(i) + 2(p(D+1) + p^2 + p(D(D+1)/2)) \quad (1)$$

where the first term represents the likelihood of the model, and the second the number of estimated parameters. $\alpha_T(i) = p(O_1 O_2 ... O_T, q_t = s_i|\lambda)$ is the probability of the partial observation sequence, $O_1 O_2 ... O_T$, until time T and state s_i at time T , given some model λ [20]. p is the number of hidden states in the HMM and D is the dimensionality of the observation vector. Figure 5 shows an example of a random HMM with four states. In the next image, we observe the AIC having a minimum in the correct number of states.

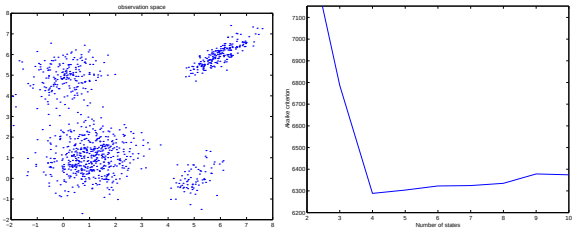

Figure 5. a) A observation space of a HMM b) AIC criteriun for order selection

3 Tracking

In this section we review the deterministic tracking of appearance models and extend it to stochastic methods using a condensation based approach.

3.1 Deterministic Tracking

Motion of planar surfaces under orthographic projection can be described in terms of affine transform (rigid) with 6 parameters $\mathbf{x}_r = (x_r^1, x_r^2, x_r^3, x_r^4, x_r^5, x_r^6)^T$:

$$\mathbf{f}(\mathbf{x}, \mathbf{x}_r) = \begin{bmatrix} x_r^1 \\ x_r^4 \end{bmatrix} + \begin{bmatrix} x_r^2 & x_r^3 \\ x_r^5 & x_r^6 \end{bmatrix} \begin{bmatrix} x - x_c \\ y - y_c \end{bmatrix} \quad (2)$$

where $\mathbf{x}_c = (x_c, y_c)^T$ is the centre position of the object template to track.

Under assumption of coplanarity, appearance tracking can be achieved by treating the region to track (N pixels) as function of affine parameters, $\mathbf{I}(\mathbf{f}(\mathbf{x}, \mathbf{x}_r)) = [I(\mathbf{f}(\mathbf{x}_1, \mathbf{x}_r)), \ldots, I(\mathbf{f}(\mathbf{x}_N, \mathbf{x}_r))]^T$. Black and Jepson [4] propose to accomplish tracking by recovering both the affine parameters $\mathbf{x}_r$ and the projection coefficients $\mathbf{c}$ by minimizing a cost function $min_{\mathbf{x}_r, \mathbf{c}} \rho[(\mathbf{I}(\mathbf{f}(\mathbf{x}, \mathbf{x}_r)) - \mathbf{U}^G \mathbf{c}), \sigma]$ where $\rho(x, \sigma)$ is a robust function (e.g. $\frac{x^2}{x^2+\sigma^2}$) to take into account violations of the model (e.g. non-coplanarity assumption, specular reflections, etc). Unlike [4] we apply Iteratively Recursive Least Squares (IRLS) to solve the robust problem in closed form and near real-time frame rates are achieved.

An extension to flexible objects can be accomplished incorporating non-rigid parameters $\mathbf{x}_{nr}$ in the warping of the image and minimizing $min_{\mathbf{x}_r, \mathbf{c}, \mathbf{x}_{nr}} \rho[\mathbf{I}\Big(\mathbf{f}(\overline{\mathbf{x}}, \mathbf{x}_r) + \sum_{k=1}^{T} x_{nr}^k \mathbf{f}(\mathbf{u}_k^S, \mathbf{x}_r)\Big) - \mathbf{U}^G \mathbf{c}, \sigma]$ [11]. Where $\mathbf{u}_k^S$ is shape mode (k column of $\mathbf{U}^S$) and $\overline{\mathbf{x}}$ is the mean shape vector.

3.2 Stochastic Tracking

Condensation [14, 17] is an algorithm which represents a tracked object's state using an entire probability distribution over the parameter space. Condensation can be seen as a particle filter for state estimation, considering the tracker as a system described by a general state space model (GSEM) where the dynamics is modeled as an autoregressive model. Any system described by GSEM can be expressed as:

$$\mathbf{x}_t = g(\mathbf{x}_{t-1}) + \eta$$
$$\mathbf{y}_t = h(\mathbf{x}_t) + \zeta$$

where $\mathbf{y}_t$ is a multidimensional vector to index the observed time series, $\mathbf{x}_t$ denotes the hidden state, η and ζ represent the noise. Without the assumption of gaussianity and linear structure in the equations, the optimal state cannot be

estimated in closed form. Condensation uses Monte Carlo methods for approximating the integrals of the posteriors by sums. Then the problem of on-line tracking is posed as one of time series analysis, where the standard filtering is performed $E[\mathbf{x}_k|\mathbf{y}_1 \dots \mathbf{y}_k]$. Given the $p(\mathbf{x}_t|\mathbf{y}_t)$ several estimators can be performed. We found the maximum a posteriori (MAP) to be more accurate than the minimum mean square error (MMSE) which produces a mixed effect of states.

3.2.1 State-space equation

Condensation assumes a linear dynamics in form of multivariate autoregressive models. These cannot model well complex motion because of the gaussian noise assumption. In order to model more complex motion, we embedded the dynamical model in the HMM structure learned in sec. 2.

Assuming that the parameters in the hidden state in the tracker ($\mathbf{x}_t = [\mathbf{x}_r^t \ \mathbf{x}_{nr}^t]^T$) are independent, the probability of the state at time t is $p(\mathbf{x}_t|\mathbf{x}_{t-1}) = p(\mathbf{x}_r^t|\mathbf{x}_r^{t-1})p(\mathbf{x}_{nr}^t|\mathbf{x}_{nr}^{t-1})$. The rigid parameters, r, are predicted with a simple constant velocity dynamical model . The shape parameters, nr, are predicted with the transition probability in the HMM $p(\mathbf{x}_{nr}^{t_i}|\mathbf{x}_{nr}^{(t-1)_j}) = a_{ij}$, where a_{ij} is the transition probability between states i and j. To provide some temporal coherence to the shape space in the sampling, we make an exponential smoothing prediction of the previous shape coefficients with those of the new sampling, that is $\mathbf{x}_{nr}^t = \alpha\mathbf{x}_{nr}^{t-1} + (1-\alpha)\mathbf{x}_{nr}^t$, provided they belong to the same state.

3.2.2 Measurement equation

The measurement equation used for tracking rigid and non-rigid motion is based on the unsupervised technique for visual learning proposed by Moghaddam et al.[19], similar to Probabilistic PCA [24], but more computational efficient. In the case of rigid and non-rigid motion the likelihood of a sample $\mathbf{x}_t$ in the state i of the HMM is computed as [19]:

$$
\begin{aligned}
p(\mathbf{y}_t|\mathbf{x}_t) &= \frac{e^{-\epsilon^2(\mathbf{x}_t)/2\eta}}{(2\pi\eta)^{(D-M)/2}} \frac{e^{-0.5\sum_i^M \frac{y_i^2}{\lambda_i}}}{(2\pi)^{\frac{M}{2}} \prod_i^M \lambda_i^{1/2}} \\
\epsilon^2(\mathbf{x}_t) &= ||\mathbf{I}_t(\mathbf{x}') - \boldsymbol{\mu}_i||_2^2 - ||(\mathbf{U}_i^G)^T(\mathbf{I}_t(\mathbf{x}') - \boldsymbol{\mu}_i)||_2^2 \\
\mathbf{x}' &= \mathbf{f}(\overline{\mathbf{x}}, \mathbf{x}_r) + [\mathbf{x}_{nr}]^T\mathbf{F}(\mathbf{U}^S, \mathbf{x}_r) \\
\mathbf{F}(\mathbf{U}^S, \mathbf{x}_r) &= [\mathbf{f}(\mathbf{u}_1^S, \mathbf{x}_r)\ \mathbf{f}(\mathbf{u}_2^S, \mathbf{x}_r)\ \cdots\ \mathbf{f}(\mathbf{u}_k^S, \mathbf{x}_r)]
\end{aligned}
\tag{3}
$$

where $\mathbf{I}_t$ is the image at time t. In fig. 6 a global description of the tracking process for the non-rigid motion is drawn. In the top figure a STM of smiling process is detailed. The smiling process has three shape states, where each ellipse represents a projection onto two first shape modes. Each shape state in the STM is represented by a $\boldsymbol{\mu}^S$ and a covariance $\boldsymbol{\Sigma}^S$, and has associated a graylevel eigenspace parameterized by $(\boldsymbol{\mu}^G, \mathbf{U}^G, \eta)$ (square). The tracking begins sampling the null state and its connected states. For each sample(non-rigid parameters), we add the rigid parameters and likelihood of this state $\mathbf{x}_t$ is evaluated with eq. 3. After, condensation is used to propagate the distribution over time. The intensity of the black in each state is proportional to the number of samples used for sampling.

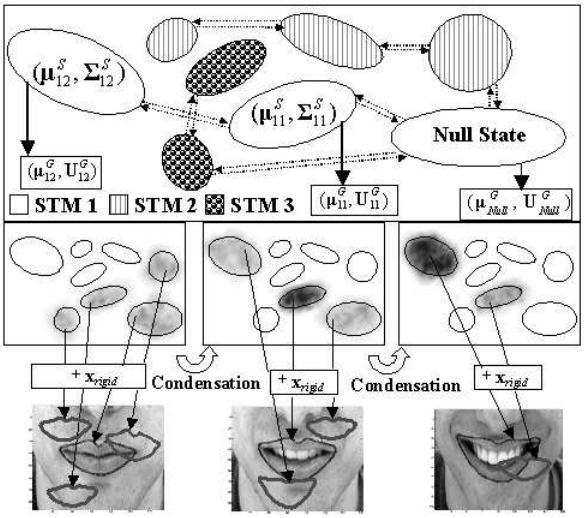

Figure 6. Spatio-temporal Tracking

In the case of rigid motion, the likelihood given the rigid parameters and the state i of the HMM will be:

$$
p(\mathbf{y}_t|\mathbf{x}_t) = \frac{1}{(2\pi)^{D/2}\prod_{k=1}^D \sigma_k} e^{-\frac{D}{2}\frac{(\mathbf{I}_t-\boldsymbol{\mu}_i)^T\mathbf{D}_i^{-1}\mathbf{W}_i(\mathbf{I}_t-\boldsymbol{\mu}_i)}{tr(\mathbf{W}_i)}} \tag{4}
$$

where $\mathbf{I}_t$ is the patch of the image at time t determined by the rigid parameters $\mathbf{x}_t$. $tr\mathbf{W}_i$ denotes trace of the diagonal matrix $\mathbf{W}_i$, and simply sums the number of pixels with spatial support equal to 1. The variance, σ_i^2, assigned to the pixels with mask 0 is the average of the variances of the pixels with mask equal to 1, in order to compare between eigenspaces with different spatial support. The extrapolation of eq. 4 to the mixture of gaussian is straightforward.

4 Experiments

In order to validate the tracking and recognition method, we have collected data of the eyes and lip-contour graylevels of 5 persons. We construct a global model of the right eye of the people for detection purposes. We also construct 5 different local spatio-temporal models, one for each person, as has been described in section 2. We perform experiments on 2D eye-tracking, out of plane eye-tracking, lip-tracking and temporal segmentation of mouth

events with the deterministic and the stochastic method proposed. The deterministic tracking is performed with visual C++ and runs on a 400Mhz standard PC at about 6 Hz from the frame grabber Matrox Meteor II (SONI EVI-D31 camera). The stochastic tracking and recognition system is implemented in unoptimized matlab code and real time rates are not achieved.

4.1 Automatic initialization and Recognition

An automatic initialization procedure is an essential part of a tracker. In this section we propose a multiresolution stochastic procedure to initialize the parametric tracker over rigid and non-rigid parameters.

At the lowest resolution level a sampling procedure is performed having a uniform prior over the whole parameter space. Similar in spirit to the strategy of simulated annealing, we propagate this distribution over scales, we sample from it while taking into account the geometric correction due to changes in scale and adding random perturbation for local search in the parameter space. Once the likelihood of each sample is calculated, we sample again from it to propagate to the next scale and so on. Note that this procedure does not correspond to a spatial filtering over scales. Fig. 7 shows the automatic initialization as a global search process in rigid and non-rigid parameter space over three scales.

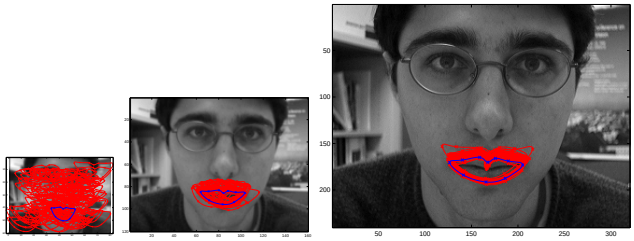

Figure 7. Initialization process at three resolution levels

We apply the same initialization procedure to eye detection. We construct a global eigenspace with all the eye's people in the database and apply the stochastic search at low resolution level. Figure 8 shows the detection of the right eye of a several persons under different illumination conditions. This method has proven to achieve good results even with global changes in illumination. Although low detection rates are produced when illumination variations are not global. We measure just the distance from feature space, since the PPCA penalize too much how far away from the mean is the sample. Once we have detected the eye, at higher resolution levels we perform the person recognition in the database, since more rich information is available. The number of samples used are 1000 and the size of the image is 160×120 at lowest resolution level.

Figure 8. Eye detection at low resolution level

4.2 2D Eye-Tracking

One of the applications explored in this paper is eye-tracking with a monocular and static camera. If just the frontal view is to be tracked this can be achieved in near real time (6 Hz) with the deterministic robust tracking using IRLS (fig. 9). We applied the tracker as a gaze detection and as a pointer to the screen. It also detects if the person's eye is close, in such case we activate an alarm to warning him.

Figure 9. Eye tracking and Gaze detection

4.3 Eye-Tracking in out of plane rotations

Tracking the eye with changes in pose becomes a difficult task not just because of the blinking of the eye, or the movement of the iris but the occlusion problem, due to the nose and the fact that the face is not a planar surface. Other difficulties arise when trying to construct a geometric model which can capture the non-linealities produced when out of plane rotations are performed. Constructing the texture manifold as explained in section 2 can avoid these problems. Figure 10 shows an eye-tracking sequence when the

person's head rotates 180 degrees. The change in the appearance of the object is due to blinking of the eye, movement of the iris, changes in pose and zoom. The tracker does not suffer from any bias in the estimation of geometric parameters with the change in appearance, since all possible combinations are represented in the mixture distribution of the HMM. The initialization procedure is automatic within our multiresolution scheme.

Figure 10. Eye-Tracking with out of plane 3D rotations and iconic changes.

Note that we track 3D but we do not calculate 3D geometrical transformations, this has been developed by Sidenbladh *et. al* [21] in the appearance based framework.

4.4 Lip-tracking

The left column of figure 13 shows a lip-tracking sequence, where non-rigid parameters are incorporated into the framework. In the top right column the sampling process in the shape space is shown, where an iso-probability contour of the gaussian cluster for each state is shown. The sampling points are marked with a cross; and we can observe how most of the points are in some cluster but other clusters have some probability depending on the HMM probabilities. The bottom right picture shows the histogram of resources assigned to each state.

4.5 Temporal segmentation of mouth events

In this section we perform the temporal segmentation of mouth events assuming that a null state occurs betweeen each pair of events. We train a HMM as explained in section 2; this HMM contains p activities (e.g. speak,smile, ...) and 1 null state which represents no activity. We perform temporal segmentation of mouth events, just applying the Viterbi algorithm [20] which gives the most likely path over the sequence. Fig. 11 shows the result of the Viterbi algorithm applied to sequence when a person is performing one of this 5 activities $1-Smile\ 2-Sad\ 3-Null\ 4-Speak\ 5-Surprise$. The x-axis is the number of frames and the y-axis is the state which goes between 1 and 5. Observe that the decoding of viterbi is necessary, otherwise if the temporal segmentation is made without imposing the temporal constraint of markovian property between states an over-segmentation is produced(fig. 11 right).

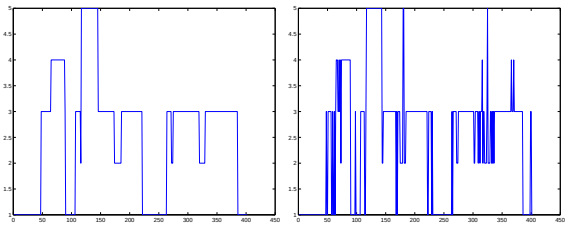

Figure 11. Temporal segmentation of mouth activities

4.6 Recognition of coupled events

In the next experiment we temporally segment sequence 13 when a person is speaking, smiling and speaking-smiling simultaneously. For mouth activity recognition, we explore the possibility of recognizing when a person is speaking and smiling simultaneously using just the learned shape/texture of speaking and smiling events. Since the physical muscles used for speaking and smiling are the same, it makes sense that when both activities are simultaneously realized there should be some correlation in their observation space. For recognition of this mixed state speak/smile, we create an artificial HMM state, where the observation space is a gaussian or mixture which covers the space between both activities. This state is generated by sampling artificially from both activities and generating linear combinations between these samples. In left figure 12 we can observe four gaussians. The 1^{st} gaussian represents the smiling process, the 2^{nd} speaking process and the $3rd$ the null state. The 4^{th} gaussian was created artificially and represents the event speaking-smiling simultaneously. Note that the artificial state speak/smile covers the space between speak and smile clusters. We include this last artificial state in the HMM for temporal segmentation and the results are shown in the right figure 12. The states represent $1-Smile\ 2-Speak\ 3-Null\ 4-Speak/Smile$. The *hybrid* state detects correctly the situations when a person is speaking and smiling simultaneously.

Acknowledgements The first author is grateful to UMI-ACS for financial support. Thanks to M. J. Black and D. Fleet for helpful comments.

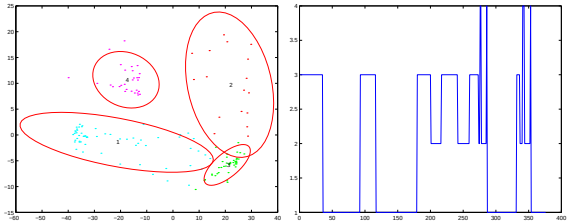

Figure 12. Generative model of speak-smile-null-speak/smile. Temporal segmentation of mouth events.

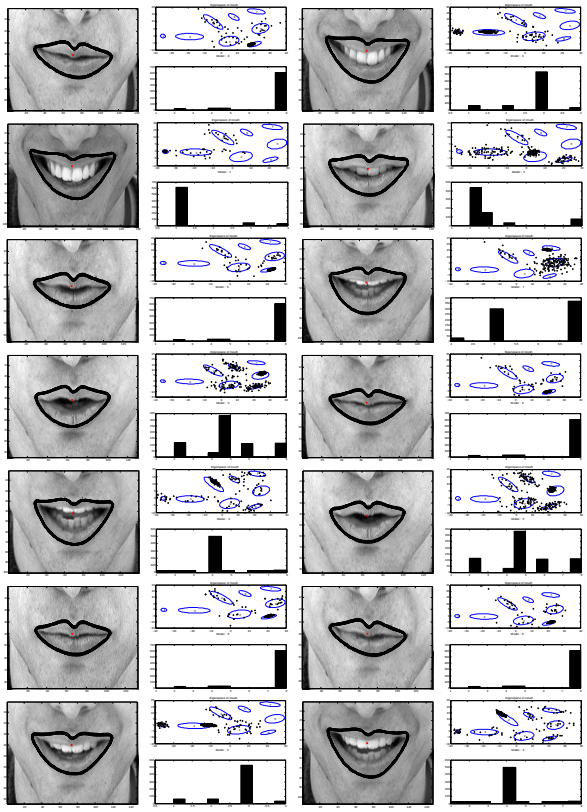

Figure 13. Tracking mouth activity when a person is speaking, smiling and speaking/smiling simultaneously.

References

[1] B. Bascle and A. Blake. Separability of pose and expression in facial track and animation. In *ICCV*, 1998.

[2] M. Black and Y. Yacoob. Recognizing facial expressions in image sequences using local parameterized models of image motion. *IJCV*, 25(1):23–48, March 1997.

[3] M. J. Black. Explaining optical flow events with parameterized spatio-temporal models. In *CVPR*, 1999.

[4] M. J. Black and A. D. Jepson. Eigentracking: Robust matching and tracking of objects using view-based representation. In *ECCV*, pages 329–342, 1996.

[5] M. Brand. Structure discovery in conditional probability models via an entropic prior and parameter extinction. In *Accepted for Neural Computation*, 1999.

[6] C. Bregler and S. Omohundro. Surface learning with applications to lipreading. In *NIPS*, 1994.

[7] M. L. Casia and S. Sclaroff. Fast,reliable tracking under varying illumination. In *CVPR*, pages 604–609, 1999.

[8] T. Cootes, G. J. Edwards, and C. Taylor. Active appearance models. In *ECCV*, 1998.

[9] T. F. Cootes, C. Taylor, D. Cooper, and J. Graham. Active shape models- their training and application. *CVIU*, 61(1):38–59, January 1995.

[10] F. de la Torre, S. Gong, and S. McKenna. View-based adaptive affine alignment. In *ECCV*, 1998.

[11] F. de la Torre, J. Vitria, P. Radeva, and J. Melenchon. Eigenfiltering for flexible eigentracking. In *Submitted to ICPR*.

[12] A. J. Heap and D. Hogg. Improving specificity in pdms using a hierarchical approach. In *BMVC*, 1997.

[13] A. J. Heap and D. Hogg. Wormholes in shape space: Tracking through discontinuous changes in shape. In *ICCV*, 1998.

[14] M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. In *ECCV*, pages 357–368, 1996.

[15] M. Isard and A. Blake. A mixed-state condensation tracker with automatic model-switching. In *ICCV*, pages 334–349, 1998.

[16] T. Jebara, K. Russell, and A. Pentland. Mixtures of eigenfeatures for real-time sturcture from texture. In *ICCV*, 1998.

[17] G. Kitagawa. Monte carlo filter and smoother for non-gaussian nonlinear state space models. *Journal of computational and Graphical Statistics*, 1(5):1–25, March 1996.

[18] G. J. McLachlan and K. Basford. *Mixture Models: Inference and Aplications to Clustering*. Marcel Dekker, Inc., 1988.

[19] B. Moghaddam and A. Pentland. Probabilistic visual learning for object detection. In *ICCV*, 1995.

[20] L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2), February 1989.

[21] H. Sidenbladh, F. de la Torre, and M. J. Black. A framework for modeling the appearance of 3d articulated figures. In *this proceedings*, 2000.

[22] L. Sirovich and M. Kirby. Low-dimensional procedure for the characterization of human faces. *J. Opt. Soc. Am. A*, 4(3):519–524, March 1987.

[23] C. Therrien. *Discrete Random Signals and Statistical Signal Processing*. Prentice Hall International, 1992.

[24] M. Tipping and C. M. Bishop. Probabilistic principal component analysis. Technical Report NCRG/97/010, Neural Computing Research Group, Aston University, 1997.

[25] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal Cognitive Neuroscience*, 3(1):71–86, 1991.

[26] A. Wilson and A. Bobick. Learning visual behavior for gesture analysis. In *ICCV*, 1995.

[27] Y. Yacoob and M. J. Black. Parameterized modeling and recognition of activities. *CVIU*, 2(73):232–247, 1999.