# Video-Based Face Recognition Using Adaptive Hidden Markov Models

Xiaoming Liu and Tsuhan Chen
Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, 15213, U.S.A.
*xiaoming@andrew.cmu.edu* *tsuhan@cmu.edu*

## Abstract

*While traditional face recognition is typically based on still images, face recognition from video sequences has become popular recently. In this paper, we propose to use adaptive Hidden Markov Models (HMM) to perform video-based face recognition. During the training process, the statistics of training video sequences of each subject, and the temporal dynamics, are learned by an HMM. During the recognition process, the temporal characteristics of the test video sequence are analyzed over time by the HMM corresponding to each subject. The likelihood scores provided by the HMMs are compared, and the highest score provides the identity of the test video sequence. Furthermore, with unsupervised learning, each HMM is adapted with the test video sequence, which results in better modeling over time. Based on extensive experiments with various databases, we show that the proposed algorithm provides better performance than using majority voting of image-based recognition results.*

## 1. Introduction

For decades human face recognition has been an active topic in the field of object recognition. A general statement of this problem can be formulated as follows: Given still or video images of a scene, identify one or more persons in the scene using a stored database of faces [1]. A lot of algorithms have been proposed to deal with the image-to-image, or image-based, recognition where both the training and test set consist of still face images. Some examples are Principal Component Analysis (PCA) [2], Linear Discriminate Analysis (LDA) [3], and Elastic Graphic Matching [4]. However, with existing approaches, the performance of face recognition is affected by different kinds of variations, for example, expression, illumination and pose. Thus, the researchers start to look at the video-to-video, or video-based recognition [5][6][7][8], where both the training and test set are video sequences containing the face.

The video-based recognition has superior advantages over the image-based recognition. First, the temporal information of faces can be utilized to facilitate the recognition task. For example, the person-specific dynamic characteristics can help the recognition [5]. Secondly, more effective representations, such as a 3D face model [9] or super-resolution images [10], can be obtained from the video sequence and used to improve recognition results. Finally, video-based recognition allows learning or updating the subject model over time. Liu et al. proposed an updating-during-recognition scheme, where the current and past frames in a video sequence are used to update the subject models to improve recognition results for future frames [8].

The temporal and motion information is a very important cue for the video-based recognition. In [5], Li suggested to model the face video as a surface in a subspace and changed the recognition problem to be a surface matching problem. Edwards et al. [6] proposed an adaptive framework on learning the human identity by using the motion information along the video sequence, which improves both face tracking and recognition. Recently, Zhou et al. proposed a probabilistic approach to video-based recognition [7]. They modeled the identity and the face motion as a joint distribution, whose marginal distribution is estimated to provide the recognition result.

The Hidden Markov Model (HMM) [11] has been successfully applied to model temporal information on applications such as speech recognition, gesture recognition [12], and expression recognition [13], etc. Samaria and Young used pixel values in each block as the observation vectors and applied HMM spatially to image-based face recognition [14]. In [15], Nefian proposed to utilize DCT coefficients as observation vectors and a spatially embedded HMM was used for recognition. Although other variations of HMM have been applied to face recognition spatially, few of them are dealing with video-based recognition.

In this paper, we apply adaptive HMM temporally to perform the video-based face recognition. As shown in Figure 1, during the training process, the statistics of training sequences, and their temporal dynamics are learned by an HMM. During the recognition process, the temporal characteristics of the test video sequence are analyzed over time by the HMM corresponding to each subject. Our proposed algorithm can learn the dynamic information and

Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'03)
1063-6919/03 $17.00 © 2003 IEEE

improve the recognition performance compared to the conventional method that simply utilizes the majority voting of image-based recognition results. Also motivated by the research in speaker adaptation [16], during the recognition process, we adapt the HMM using the test sequences. Thus an updated HMM can provide better modeling and result in better performance over time.

The paper is organized as follows. In the next section, we briefly introduce HMM. In Section 3 our algorithms will be presented in detail. We discuss how to adapt the HMM in order to enhance the modeling and recognition performance. In Section 4, we compare the recognition performance of our algorithm with a baseline algorithm applied to various databases. Finally this paper is concluded in Section 5.

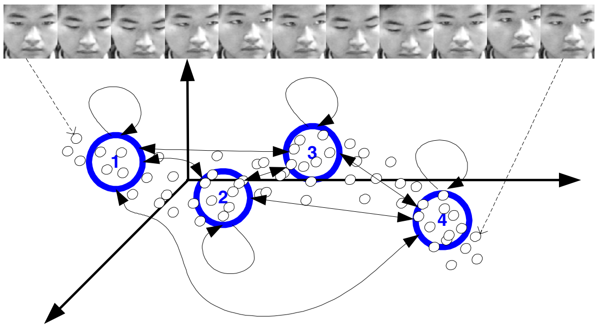

**Figure 1 Temporal HMM for modeling face sequences**

## 2. Hidden Markov model

A Hidden Markov Model is a statistical model used to characterize the statistical properties of a signal [11]. An HMM consists of two stochastic processes: one is an unobservable Markov chain with a finite number of states, an initial state probability distribution and a state transition probability matrix; the other is a set of probability density functions associated with each state. There are two types of HMM: discrete HMM and continuous HMM. The continuous HMM is characterized by the following:

- $N$, the number of states in the model. We denote the individual state as $\mathbf{S} = \{S_1, S_2, \cdots, S_N\}$, and the state at time $t$ as $q_t$, $1 \le t \le T$, where $T$ is the length of the observation sequence.
- **A**, the state transition probability matrix, i.e., $\mathbf{A} = \{a_{ij}\}$, where
$$a_{ij} = P[q_t = S_j \mid q_{t-1} = S_i], \qquad 1 \le i, j \le N$$
with the constrain,
$$\sum_{j=1}^{N} a_{ij} = 1, \qquad 1 \le i \le N \, .$$
- **B**, the observation probability density functions (pdf), i.e., $\mathbf{B} = \{b_i(\mathbf{O})\}$, where
$$b_i(\mathbf{O}) = \sum_{k=1}^{M} c_{ik} N(\mathbf{O}; \boldsymbol{\mu}_{ik}, \mathbf{U}_{ik}), \qquad 1 \le i \le N \qquad (1)$$
where $c_{ik}$ is the mixture coefficient for $k^{\text{th}}$ mixture component in State $i$. $M$ is the number of components in a Gaussian mixture model. $N(\mathbf{O}; \boldsymbol{\mu}_{ik}, \mathbf{U}_{ik})$ is a Gaussian pdf with the mean vector $\boldsymbol{\mu}_{ik}$ and the covariance matrix $\mathbf{U}_{ik}$.
- $\boldsymbol{\pi}$, the initial state distribution, i.e., $\boldsymbol{\pi} = \{\pi_i\}$, where $\pi_i = P[q_1 = S_i]$, $\quad 1 \le i \le N$ .

Using a shorthand notation, an HMM is defined as the triplet
$$\boldsymbol{\lambda} = (\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}) .$$

## 3. Our proposed algorithms

In this section, we describe our proposed algorithms in detail. First, feature extraction, HMM training and HMM testing are presented. Then, we introduce an algorithm to adapt the HMM in order to enhance the modeling over time.

### 3.1 Temporal HMM

When applying HMM to face recognition, researchers have proposed to use different features, for example, pixels values [14], eigen-coefficients and DCT coefficients [15], as the observation vectors. In our algorithm, each frame in the video sequence is considered as one observation. Since PCA gives the optimal representation of the images in terms of the mean square error, all face images are reduced to low-dimensional feature vectors by PCA. Given a face database with $L$ subjects and each subject has a training video sequence containing $T$ images.
$$\mathbf{F}_l = \{\mathbf{f}_{l,1}, \mathbf{f}_{l,2}, \cdots, \mathbf{f}_{l,T}\} \qquad 1 \le l \le L$$

Each image only contains the face portion. By performing eigen-analysis for these $L \times T$ samples, we obtain an eigenspace with a mean vector $\mathbf{m}$ and a few eigenvectors $\{\mathbf{V_1}, \mathbf{V_2}, \cdots, \mathbf{V_d}\}$. All the training images are projected into this eigenspace and generate corresponding feature vectors, $\mathbf{e}_{lt}, 1 \le l \le L, 1 \le t \le T$ , which will be used as observation vectors in the HMM training. At this stage, we also compute the covariance matrix of all the feature vectors $\mathbf{e}_{lt}$ , $\mathbf{C}_e$ , which is a diagonal matrix with eigenvalues as the diagonal elements. The matrix $\mathbf{C}_e$ describes in general how all the face images distribute on each dimension of a low-dimensional eigenspace, which provides useful information for the HMM training. Essentially in our algorithm, PCA is used for the dimension reduction purpose.

Each subject in the database is modeled by a $N$-state fully connected HMM. The feature vectors $\mathbf{e}_{lt}, 1 \le t \le T$

form the observation vectors $\mathbf{O}$ for training the HMM of Subject $l$. The training for each HMM is as follows. First, the HMM $\boldsymbol{\lambda} = (\mathbf{A}, \mathbf{B}, \boldsymbol{\pi})$ is initialized. Vector quantization is used to separate the observation vectors into $N$ classes and the observation vectors associated with each class are used to generate the initial estimates for **B,** i.e., estimate $c_{ik}$ , $\boldsymbol{\mu}_{ik}$ and $\mathbf{U}_{ik}$ as in (1). Second, in order to maximize the likelihood $P(\mathbf{O} | \boldsymbol{\lambda})$, the model parameters are re-estimated by using the Expectation Maximization (EM) algorithm [16]. It produces a sequence of estimates for $\boldsymbol{\lambda}$ , given a set of observation vectors $\mathbf{O}$ , so that each estimate $\boldsymbol{\lambda}^{(n)} = (\mathbf{A}, \mathbf{B}, \boldsymbol{\pi})$ has a larger value of $P(\mathbf{O} | \boldsymbol{\lambda})$ than the preceding estimate $\boldsymbol{\lambda}^{(n-1)}$ . The re-estimation is defined as follows:

$$\pi_i = \frac{P(\mathbf{O}, q_1 = i \mid \boldsymbol{\lambda})}{P(\mathbf{O} \mid \boldsymbol{\lambda})} \tag{2}$$

$$a_{ij} = \frac{\sum_{t=1}^{T} P(\mathbf{O}, q_{t-1} = i, q_t = j \mid \boldsymbol{\lambda})}{\sum_{t=1}^{T} P(\mathbf{O}, q_{t-1} = i \mid \boldsymbol{\lambda})} \tag{3}$$

$$c_{ik} = \frac{\sum_{t=1}^{T} P(q_t = i, m_{q_t t} = k \mid \mathbf{O}, \boldsymbol{\lambda})}{\sum_{t=1}^{T} \sum_{k=1}^{M} P(q_t = i, m_{q_t t} = k \mid \mathbf{O}, \boldsymbol{\lambda})} \tag{4}$$

$$\boldsymbol{\mu}_{ik} = \frac{\sum_{t=1}^{T} \mathbf{O}_t P(q_t = i, m_{q_t t} = k \mid \mathbf{O}, \boldsymbol{\lambda})}{\sum_{t=1}^{T} P(q_t = i, m_{q_t t} = k \mid \mathbf{O}, \boldsymbol{\lambda})} \tag{5}$$

$$\mathbf{U}_{ik} = (1-\alpha)\mathbf{C}_e + \alpha \frac{\sum_{t=1}^{T} (\mathbf{O}_t - \boldsymbol{\mu}_{ik})(\mathbf{O}_t - \boldsymbol{\mu}_{ik})^T P(q_t = i, m_{q_t t} = k \mid \mathbf{O}, \boldsymbol{\lambda})}{\sum_{t=1}^{T} P(q_t = i, m_{q_t t} = k \mid \mathbf{O}, \boldsymbol{\lambda})} \tag{6}$$

where $m_{q_t t}$ indicates the mixture component for State $q_t$ and time $t$. Equ. (6) is used to adapt the variance estimate from $\mathbf{C}_e$, which is a general model for the variance of all subjects. The parameter $\alpha$ is a weighting factor and is chosen as 0.5 in our experiments. Normally during the training process, when the number of face images assigned to each state is less than the dimension of feature vectors, $\mathbf{U}_{ik}$ will be a singular matrix. This adaptation step can prevent this from happening. The model parameters are estimated iteratively using (2)-(6) until the likelihood $P(\mathbf{O} | \boldsymbol{\lambda})$ converges.

In the recognition process, given a video sequence containing face images, all frames are projected into the eigenspace and the resulting feature vectors form the observation vectors. Then the likelihood score of the observation vectors given each HMM is computed, where the transition probability and observation pdf are used. The face sequence is recognized as Subject $k$ if:

$$P(\mathbf{O} \mid \boldsymbol{\lambda}_k) = \max_l P(\mathbf{O} \mid \boldsymbol{\lambda}_l)$$

### 3.2 Adaptive HMM

In typical speech recognition systems, there is a dichotomy between speaker-independent and speaker-dependent systems. While speaker-independent systems are ready to be used without further training, their performance is usually two or three times worse than that of speaker-dependent systems. However, the latter requires large amounts of training data from the designated speaker. To address this issue, the concept of speaker adaptation [16][17] has been introduced, where a small amount of data from the specific speaker are used to modify the speaker-independent system and improve its performance. Similarly, in the vision community, Liu et al. [8] also proposed unsupervised model updating to enhance the object modeling and improve the recognition performance over time.

Motivated by these ideas, we propose to use adaptive HMM for video-based face recognition. That is, during the recognition process, after we recognized one test sequence as one subject, we can use this sequence to update the HMM of that subject, which will learn the new appearance in this sequence and provide an enhanced model of that subject. Obvious two questions need to be answered. First, how do we decide whether we should use the current test sequence for updating? This is important for avoiding wrong updating, i.e., one sequence is used to update other subject's model, instead of his/her own model. Second, how do we adapt HMM?

Essentially for the first question, we would like to measure how confident that the recognition result for the current sequence is correct based on a certain feature. The more the confidence, the more certain we should use the current sequence to update the HMM. In our algorithm, we use the *likelihood difference*, i.e., the difference between the highest likelihood score and the second highest likelihood score, as the feature to make the decision. The reason is that for correct recognition, the likelihood difference tends to be large; while for incorrect recognition, the likelihood difference is usually small. So given a test sequence, we compare its likelihood difference with a pre-defined threshold, and update the HMM only if the likelihood difference is larger than the threshold. In practice, this pre-defined threshold can be determined by performing experiments on a cross-validation data set.

We use the standard MAP adaptation technique [16] to adapt the HMM. That is, given an existing HMM $\boldsymbol{\lambda}^{old}$ and observation vectors $\mathbf{O}$ from a test sequence, we estimate a new HMM, $\boldsymbol{\lambda} = (\mathbf{A}, \mathbf{B}, \boldsymbol{\pi})$ . We use $\boldsymbol{\lambda}^{old}$ as the initial parameters of $\boldsymbol{\lambda}$, and the EM algorithm is used to re-estimate $\boldsymbol{\lambda}$ except that the mean estimation is as follows:

$$\boldsymbol{\mu}_{ik} = (1-\beta)\boldsymbol{\mu}_{ik}^{old} + \beta \frac{\sum_{t=1}^{T} \mathbf{O}_t P(q_t = i, m_{q_t t} = k \mid \mathbf{O}, \boldsymbol{\lambda})}{\sum_{t=1}^{T} P(q_t = i, m_{q_t t} = k \mid \mathbf{O}, \boldsymbol{\lambda})} \quad (7)$$

where $\boldsymbol{\mu}_{ik}^{old}$ is the mean vector from the existing HMM $\boldsymbol{\lambda}^{old}$ and $\beta$ is a weighting factor that gives the bias between the previous estimate and the current data. In our experiments, we choose $\beta$ to be 0.3. Also during the iteration of the EM algorithm, we do not update the covariance matrix of the observation pdf, $\mathbf{U}_{ik}$, because from speech research literature, the major discriminative information of an HMM is retained by the mean vectors instead of the covariance matrixes. Also in our experiments, updating the covariance matrix does not show significant improvements compared to no updating.

# 4. Experiments

## 4.1 Setup

In order to test the proposed algorithm, we have collected a Task database with 21 subjects. During the data collection, each subject is required to read a paper that is hung beside the monitor and type it using the keyboard. Thus essentially the subject switches between reading the paper, looking at the monitor and looking at the keyboard. For each subject we collected 2 sequences, where one has 322 frames and is used for training; the other has around 400 frames and is used for testing. From the whole video frame, we manually crop the face region as a face image with 16 by 16 pixels. Sample face images for some subjects are shown in Figure 2. In addition, 5 months after we captured the original Task database, we captured a new test set with different lighting conditions and camera settings, but with only 11 subjects available.

The second database is the Mobo database [18], originally collected for human identification from distance. There are 24 subjects in this database. Each subject has four sequences captured in different walking situations: holding a ball, fast walking, slow walking, and walking on the incline. Each sequence has 300 frames. Three frames from one sequence are shown in Figure 3. Large head pose variation can be observed from this database. We crop the face portion from each frame and use it for experiments. The image size is 48 by 48 pixels. Some of the manually cropped faces are shown in Figure **4**. For each subject, the first 150 frames of all four sequences are used for training, and the remaining 150 frames of all sequences are used for testing.

In order to mimic the practical situation, we use the test scheme shown in Figure 5. That is, given a test set with $L$ subjects, we randomly choose a subject $l$, a starting frame $k$ and a length $z$. Then frames $\{\mathbf{f}_{l,k}, \mathbf{f}_{l,k+1}, \cdots \mathbf{f}_{l,k+z-1}\}$ form a sequence for testing. Essentially this is similar to the practical situation where any subject can come to the recognition system at any time and with any duration. For both databases, we use this scheme to create a large amount of sequences for testing.

**Figure 2 Sample face images from our Task database**

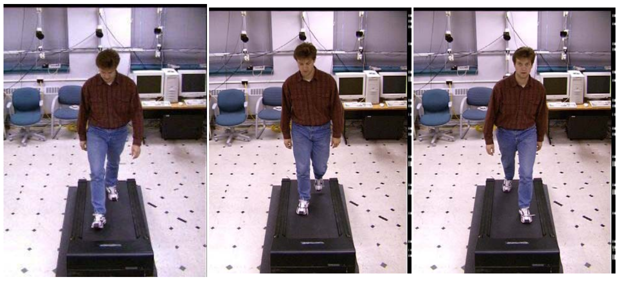

**Figure 3 Sample images from the Mobo database**

**Figure 4 Cropped faces from the Mobo database**

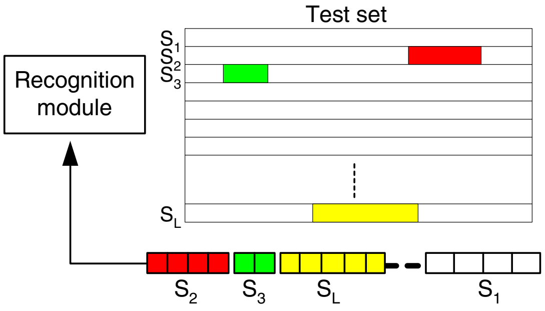

**Figure 5 Test scheme**

## 4.2 Experimental results

In order to compare video-based recognition with image-based recognition, we choose the individual PCA method (IPCA) as a baseline image-based algorithm. It has been a popular face recognition method to build an individual eigenspace for each subject and perform recognition based on the residue [3]. Given a face sequence, after applying IPCA recognition to each frame, majority voting determines the identification of the whole sequence. We use this method as the baseline algorithm.

With the test scheme in Figure 5, we test the baseline algorithm on the Task database using 1000 sequences. By choosing different numbers of eigenvectors for the baseline algorithm, we can obtain different recognition performance. Typically the more eigenvectors the baseline algorithm uses, the better performance it has. However, after a certain number of eigenvectors are used, the performance does not improve any further. For the Task database, when 12 eigenvectors are used, the baseline algorithm has the best performance: 9.9% recognition error rate. We also apply our algorithm on this database with the same 1000 sequences. We use 45 eigenvectors during the dimension reduction and a 12-state HMM is used to model each subject. Generally speaking, the more states used to train one HMM, the better modeling we have, while we also have more parameters to estimate. For the Task database, we found 12 states is a good compromise between modeling and estimation. Each state uses one Gaussian distribution to model the observation probability. Eventually we obtain 7.0% recognition error rate, which is better than the best result we can get from the baseline algorithm. We also apply the adaptive HMM on the same test set and obtain 4.0% recognition error rate. Also, we do the same comparison with the newly captured test set. As shown in Table 1, although the overall recognition rate is a lot higher than the first test set because of the time elapse and the lighting and camera variations, we still see that our proposed methods work better than the baseline algorithm. However, the adaptive HMM does not improve a lot comparing to the temporal HMM because when the overall recognition error rate goes high, it is more likely to make wrong updating.

Similarly we apply these three algorithms to the Mobo database as well. The same test scheme is utilized and 500 randomly chosen sequences are used for testing. Different numbers of eigenvectors have been used for the baseline algorithm. It has the best performance with 2.4% recognition error rate when 7 eigenvectors are used. For our temporal HMM, we use 30 eigenvectors during the dimension reduction and train a 14-state HMM for each subject, where each state has one Gaussian distribution for modeling the observation probability. The recognition error rate is 1.6% for the temporal HMM. Similarly we also apply the adaptive HMM for the Mobo database and obtain 1.2% recognition error rate.

We summarize the performance comparison among three algorithms in Table 1. As we can see, in both databases, our proposed algorithms perform much better than the baseline algorithm. Especially, the adaptive HMM algorithm almost halves the error rate of the baseline algorithm.

**Table 1 Comparison among three algorithms**

| | Baseline | Temporal HMM | Adaptive HMM |
|---|---|---|---|
| Mobo | 2.4% | 1.6% | 1.2% |
| Task | 9.9% | 7.0% | 4.0% |
| Task-new set | 49.1% | 31.0% | 29.8% |

There are a few reasons why the proposed algorithms work better than the baseline algorithm. The first is that HMM is able to learn both the dynamics and the temporal information. The second is that there is mismatching between the training and test sets, i.e., some of the test sequences show the new appearance that is barely seen in the training set. So the adaptive HMM enables the HMM to learn this new appearance in the test set and thus enhance the modeling. The third is the modeling ability of using observation pdf corresponding different states. For example, Figure 6 shows the training images of one subject in a subspace composed by the first three eigenvectors, $\{\mathbf{V}_1, \mathbf{V}_2, \mathbf{V}_3\}$. The plus signs show the feature vectors of all images from one training sequence. It illustrates that it is hard for IPCA to model this arbitrary distribution effectively since IPCA essentially assumes a single Gaussian distribution, while the four dots, which are the means of observation pdf corresponding to four states, can model the whole distribution better. We also plot the four means as images in Figure 7, where they seem to represent different head poses in the training sequence.

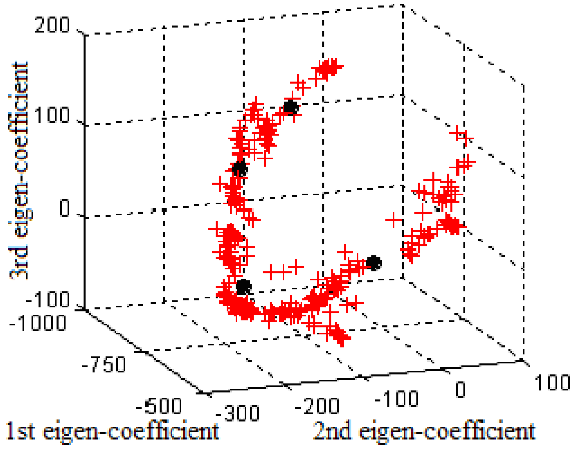

**Figure 6 The distribution of training faces in the eigenspace**

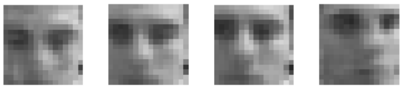

**Figure 7 The mean faces corresponding to four states**

## 5. Conclusions

In this paper, we propose to use adaptive HMM to perform video-based face recognition. During the training process, the statistics of training video sequences of each subject, and their temporal dynamics are learned by an HMM. During the recognition process, the temporal characteristics of the test video sequence are analyzed over time by the HMM corresponding to each subject. The likelihood scores provided by the HMMs are compared, and the highest score provides the identity of the test video sequence. Furthermore, with unsupervised learning, each HMM is adapted with the test video sequence, which results in better modeling over time. Based on extensive experiments with various databases, we show that the proposed algorithm provides better performance than using majority voting of image-based recognition results.

The paper shows that video-based face recognition is one promising way to enhance the performance of current image-based recognition. Along this direction, our future work is to combine the idea of spatial HMM with our temporal HMM to model both spatial and temporal information of the face sequences. Also, since the observation probabilities of HMM is used to model facial appearance, we can utilize it for face tracking, which enables both face tracking and recognition in the same framework.

## Acknowledgment

The authors would like to thank Dr. J. Shi, R. Gross, Carnegie Mellon University, and Prof. V. Krueger, Aalborg University, for providing the Mobo database for our experiments. The authors also acknowledge each individual appearing in our face database.

## References

[1] R. Chellappa, C.L. Wilson, and S. Sirohey, "Human and machine recognition of faces: a survey", *Proceedings of the IEEE*, Vol.83, No.5, 1995, pp.705–741.

[2] M. Turk and A. Pentland, "Eigenfaces for Recognition", *Journal of Cognitive Neuroscience*, Vol.3, No.1, 1991, pp.71-86.

[3] P.N. Belhumeur, J.P. Hespanha, and D.J. Kriegman, "Eiegnfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol.19, No.7, 1997, pp.711-720.

[4] M. Lades, J.C. Vorbruggen, J. Buhmann, J. Lange, C. von der Malsburg, R.P. Wurtz, and W. Konen, "Distortion Invariant Object Recognition in the Dynamic Link Architecture", *IEEE Transactions on Computers*, Vol.42, No.3, 1992, pp.300-311.

[5] Y. Li, *Dynamic face models: construction and applications*, PhD Thesis, Queen Mary, University of London, 2001.

[6] G. J. Edwards, C.J. Taylor, T.F. Cootes, "Improving Identification Performance by Integrating Evidence from Sequences", *In Proc. Of 1999 IEEE Conference on Computer Vision and Pattern Recognition,* June 23-25, 1999 Fort Collins, Colorado, pp.486-491.

[7] S. Zhou, V. Krueger, and R. Chellappa, "Face Recognition from Video: A CONDENSATION Approach", In *Proc. of Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, Washington D.C., May 20-21, 2002, pp.221-228.

[8] X. Liu, T. Chen and S. M. Thornton, "Eigenspace Updating for Non-Stationary Process and Its Application to Face Recognition", To appear in *Pattern Recognition*, Special issue on Kernel and Subspace Methods for Computer Vision, September 2002.

[9] A. Roy Chowdhury, R. Chellappa, R. Krishnamurthy and T.Vo, "3D Face Recostruction from Video Using A Generic Model", In *Proc. of Int. Conf. on Multimedia and Expo*, Lausanne, Switzerland, August 26-29, 2002.

[10] S. Baker and T. Kanade, "Limits on Super-Resolution and How to Break Them", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 9, September 2002, pp.1167-1183.

[11] L. Rabiner, "A tutorial on Hidden Markov Models and selected applications in speech recognition", *Proceedings of the IEEE*, Vol.77, No.2, 1989, pp.257-286.

[12] A. Kale, A.N. Rajagopalan, N. Cuntoor and V. Krueger, "Gait-based Recognition of humans Using Continuous HMMs*", In proceedings of the 5th IEEE International Conference on Automatic Face and Gesture Recognition*, Washinton D.C. May 20-21, 2002, pp.336-341.

[13] J.J. Lien, *Automatic Recognition of Facial Expressions Using Hidden Markov Models and Estimation of Expression Intensity*, doctoral dissertation, tech. report CMU-RI-TR-98-31, Robotics Institute, Carnegie Mellon University, April 1998.

[14] F. Samaria and S. Young, "HMM-based architecture for face identification", *Image and vision computing*, Vol.12, No.8, Oct 1994.

[15] A. Nefian, *A hidden Markov model-based approach for face detection and recognition*, PhD thesis, Georgia Institute of Technology, Atlanta, GA. 1999.

[16] J-L. Gauvain and C-H. Lee, "Maximum a Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains", *IEEE Transactions on Speech and Audio Processing*, Vol.2, No.2, 1994, pp.291-298.

[17] C. J. Leggetter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of the parameters of continuous density hidden markov models", *Computer Speech and Language*, Vol.9, 1995, pp. 171-185.

[18] R. Gross and J. Shi, The CMU Motion of Body (MoBo) Database, tech. report CMU-RI-TR-01-18, Robotics Institute, Carnegie Mellon University, June, 2001.