FACE RECOGNITION USING MIXTURES OF PRINCIPAL COMPONENTS

Deepak S. Turaga and Tsuhan Chen

Video and Display Processing
Philips Research USA
Briarcliff Manor, NY 10510
deepak.turaga@philips.com

Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA 15213
tsuhan@cmu.edu

ABSTRACT

We introduce an efficient statistical modeling technique called Mixture of Principal Components (MPC). This model is a linear extension to the traditional Principal Component Analysis (PCA) and uses a mixture of eigenspaces to capture data variations. We use the model to capture face appearance variations due to pose and lighting changes. We show that this more efficient modeling leads to improved face recognition performance.

1. INTRODUCTION

Face recognition has generated much interest in the research community primarily because of the multitude of applications it enables. Automatic face recognition is very useful as a non-intrusive authentication, verification and identification tool. Among face recognition techniques template matching techniques are very popular. Template matching involves building a template or model for an object in the database and then using that to classify the test face. A sample template matching recognition system is shown in Figure 1.

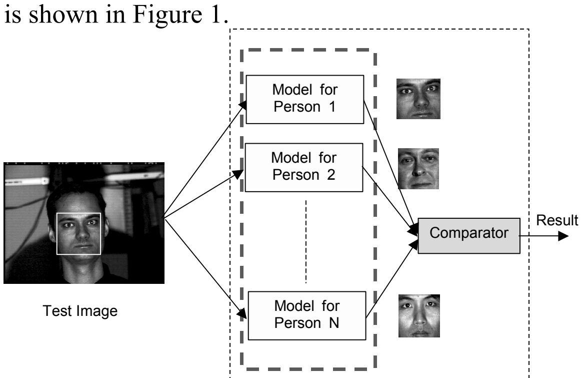

Figure 1. Sample face recognition system

In Figure 1 the test face is matched with the models in the database, each of which returns a score that may be in terms of probabilities, likelihoods, distances etc. These scores are then passed to a comparator to make the final decision. The focus of this paper is on improving face recognition performance using better models for the faces.

The eigenface approach for recognition was proposed by Turk and Pentland [1] who used principal component analysis (PCA) [2] to create an eigenspace for all the subjects in the database. The test face is projected onto this eigenspace and the resulting coefficients are used to classify it among one of the many subjects.

We first examine the PCA as a model for face recognition and find that it is inefficient at capturing data with large amounts of variation. For instance, when the data consists of multiple clusters, a mixture of eigenspaces is more efficient at capturing the data variations. We thus examine many non-linear and linear extensions to the PCA for improved modeling efficiency. Among the non-linear extensions is the work by Hastie and Stuetzle [3], who proposed principal surfaces as an alternative to PCA. This involves modeling the data clusters using parameterized surfaces instead of the hyperplanes that PCA uses. Many neural network approximators for these principal surfaces of the high-dimensional data have also been proposed. Among these are the work by Oja [4] and by Kung and Diamantaras [5]. Other non-linear techniques such as Multi-Dimensional Scaling (MDS) [6] have also been introduced. Recently, other similar approaches to dimensionality reduction such as Locally Linear Embedding (LLE) [7] have also been proposed. LLE attempts to preserve local relationships between data points during dimensionality reduction. However all these non-linear techniques are computationally intensive and also lack an easy forward-backward transformation.

We then examine other linear extensions to the PCA. Among these extensions is the Vector Quantization PCA (VQPCA) [8]. This technique modifies the traditional VQ algorithm by changing the optimization criterion to include reconstruction error. Data samples are partitioned into clusters based on which cluster reconstructs them with smallest error. The parameters of each cluster are then updated using local PCAs and this process is iterated till convergence of parameters. This hard partitioning of data into clusters before dimensionality reduction leads to loss of the global information present in the data. We would like to both exploit the local as well as global information present in the data and so prefer a soft partitioning of the data. The idea of soft partitioning the data while training local PCAs has been examined by Tipping and Bishop [9]. They first introduce an extension to the PCA called the Probabilistic PCA (PPCA) and use a mixture of such PPCAs to represent the data. However, during this dimensionality reduction, the squared error is not explicitly minimized; instead the likelihood of observing the data given the model is maximized. This may lead to poor reconstruction performance of the model.

We propose a linear extension to the PCA called *Mixture of Principal Components (MPC)*. Similar to the way that a Gaussian mixture models the data distribution, the MPC automatically models the data using a mixture of eigenspaces. However, instead of optimizing the likelihood of observing the data given the model, the MPC parameters are chosen to minimize the overall reconstruction error. It is efficient, accurate and the reconstruction is easy to compute. We hence use the MPC to model the faces and improve recognition performance.
This paper is organized as follows. Section 2 describes the new statistical modeling technique, mixtures of principal components (MPC). Section 3 describes the test database and includes face recognition results. We conclude in Section 4 and indicate directions for future research.

2. MIXTURES OF PRINCIPAL COMPONENTS

The MPC model for any data set is characterized by two sets of parameters, the means and the eigenvectors of the component eigenspaces. In order to describe our model more efficiently, we use the following notation.

N : Number of training data vectors

D : Dimension of data vectors

M : Number of mixture components

P : Number of eigenvectors in each mixture component

$\mathbf{x}_i$: Training Data vectors $(i = 1 \ldots N)$

$\hat{\mathbf{x}}_{ij}$: Reconstruction for $\mathbf{x}_i$ from component j $(j = 1 \ldots M)$

$\mathbf{y}_i$: Test vector i

$\hat{\mathbf{y}}_{ij}$: Reconstruction for $\mathbf{y}_i$ from component j $(j = 1 \ldots M)$

$\hat{\mathbf{Y}}_i$: Matrix with $\hat{\mathbf{y}}_{ij}$ as columns $(D \times M)$

$\mathbf{m}_j$: Mean of the j^{th} mixture component

$\mathbf{u}_{jk}$: k^{th} eigenvector of j^{th} mixture component

$\mathbf{U}_j$: Eigenvector matrix for j^{th} mixture component $(D \times P)$

$\mathbf{w}_i$: Weight vector to combine M reconstructions for vector i. It has elements w_{ij} $(j = 1 \ldots M)$

2.1. Reconstruction from Model

Our approach to reconstruction consists of linearly combining individual reconstructions from a mixture of component eigenspaces. Given a data test vector $\mathbf{y}_i$, we first project it onto each of the component eigenspaces to obtain individual reconstructions $\hat{\mathbf{y}}_{ij}$ and then linearly combine these individual reconstructions to obtain the representation that is closest to the original data vector $\mathbf{y}_i$. The individual reconstruction $\hat{\mathbf{y}}_{ij}$ for test vector i from mixture component j is obtained as shown in (1).

$$\hat{\mathbf{y}}_{ij} = \mathbf{m}_j + \sum_{k=1}^{P} \left[(\mathbf{y}_i - \mathbf{m}_j)^T \mathbf{u}_{jk} \right] \mathbf{u}_{jk} \qquad (1)$$

These individual reconstructions are then linearly combined using a set of weights. We show an illustration of our approach in Figure 2.

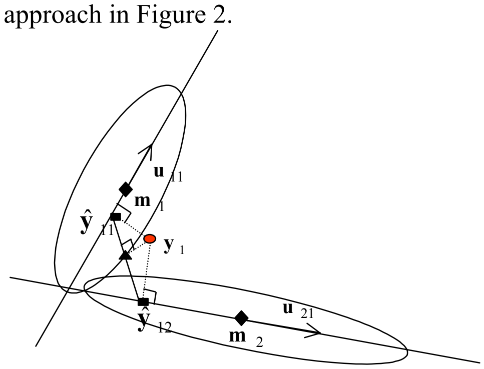

Figure 2. Illustration of mixture of eigenspaces

In Figure 2 we show data reconstruction using a mixture of two component eigenspaces, each with one eigenvector. The component eigenspaces have means $\mathbf{m}_1$ and $\mathbf{m}_2$, and eigenvectors $\mathbf{u}_{11}$ and $\mathbf{u}_{21}$ respectively. In the figure, the means are shown as black diamonds and the direction of the eigenvectors is shown as a line passing through the corresponding means. Given a data sample $\mathbf{y}_1$, shown as a dark circle, we first project it onto each of the component eigenspaces to obtain $\hat{\mathbf{y}}_{11}$ and $\hat{\mathbf{y}}_{12}$. We then linearly combine these two projections to obtain the best reconstruction for the data, shown as the dark triangle in the figure. Due to the nature of the linear weighting, the best combination lies along the line joining the two individual reconstructions. The weights are chosen so that the resulting combination is as close to the data sample as possible, i.e., it lies on the perpendicular from the data sample to the line joining the two individual reconstructions.

The weights are solved for individually for each of the test vectors. The only constraint that we impose on the weights is that they are required to sum to one.

2.2. Training of Model Parameters

This section focuses on determining the mixture means and eigenvectors given a set of training data. We would like to automatically train the MPC to minimize the reconstruction error. We formulate this training problem as a minimum error optimization problem and provide a solution using an iterative Expectation Maximization (EM) kind of algorithm.

Given a set of N training data vectors, we model them using an MPC containing M eigenspaces, each with P eigenvectors. This problem of minimizing the squared error through the choice of the means and the sets of eigenvectors may be mathematically written as in (2).

$$\min_{\mathbf{m}_j,\mathbf{u}_{jk}} \frac{1}{N}\sum_{i=1}^{N}\left\| \mathbf{x}_i - \sum_{j=1}^{M} w_{ij}\left[\underbrace{\mathbf{m}_j + \sum_{k=1}^{P}\left[\left(\mathbf{x}_i - \mathbf{m}_j\right)^T \mathbf{u}_{jk}\right]\mathbf{u}_{jk}}_{\hat{\mathbf{x}}_{ij}}\right]\right\|^2 \quad (2)$$

The weights are recomputed for each of the training data vectors, and are used to update the means and the eigenvectors, but are not part of the model.

We first initialize the means and the eigenvectors for the different components randomly. We compute the weights for each data vector, following which we update the means, while keeping the eigenvectors fixed and then use the new means to update the set of eigenvectors. After this we recompute the weights and repeat the update procedure till convergence.

We include the update equations for the means and eigenvectors; the derivation is presented in [10].
The mean of mixture component q is updated as in (3).

$$\mathbf{m}_q = \frac{1}{\sum_{i=1}^{N} w_{iq}^2}\left[\sum_{i=1}^{N} w_{iq}\left(\mathbf{x}_i - \sum_{j=1, j\neq q}^{M} w_{ij}\hat{\mathbf{x}}_{ij}\right)\right] \quad (3)$$

The eigenvectors for the r-th mixture component are obtained as eigenvectors of the matrix $\mathbf{C}_r$ defined in (4).

$$\frac{1}{N}\sum_{i=1}^{N}\left[\begin{array}{l} w_{ir}\left[\left(\mathbf{x}_i - \mathbf{m}_r\right)\mathbf{x}_i^T + \mathbf{x}_i\left(\mathbf{x}_i - \mathbf{m}_r\right)^T\right] - \\ \sum_{j=1}^{M} w_{ij} w_{ir}\left[\left(\mathbf{x}_i - \mathbf{m}_r\right)\mathbf{m}_j^T + \mathbf{m}_j\left(\mathbf{x}_i - \mathbf{m}_r\right)^T\right] - \\ \sum_{j=1, j\neq r}^{M}\sum_{k=1}^{P} w_{ij} w_{ir}\left[\begin{array}{l}\left[\mathbf{u}_{jk}^T\left(\mathbf{x}_i - \mathbf{m}_j\right)\right]\times \\ \left[\left(\mathbf{x}_i - \mathbf{m}_r\right)\mathbf{u}_{jk}^T + \mathbf{u}_{jk}\left(\mathbf{x}_i - \mathbf{m}_r\right)^T\right]\end{array}\right] - \\ w_{ir}^2\left(\mathbf{x}_i - \mathbf{m}_r\right)\left(\mathbf{x}_i - \mathbf{m}_r\right)^T \end{array}\right] \quad (4)$$

The first P eigenvectors of this matrix are the desired eigenvectors of the mixture component.

3. FACE RECOGNITION RESULTS

In order to test modeling performance, we use faces from the PIE database [11]. The database contains faces with multiple poses, under different illuminations and with different expressions. We use faces with extreme pose and illumination variations as our test set. As an illustration of the variations in the database, we show images from one of the subjects in Figure 3.

Figure 3. Sample pose and illuminations in database

As can be seen from Figure 3, the pose variations include views from left to right as well from the top and the bottom. Besides these pose variations, the lighting also varies from bright to extremely dark.

We use five persons from the database with 286 images per person. We use half of these to train the models and test recognition performance on the other half. We train mixtures with four components and two eigenvectors each. One set of parameters for a person in the database is shown in Figure 4.

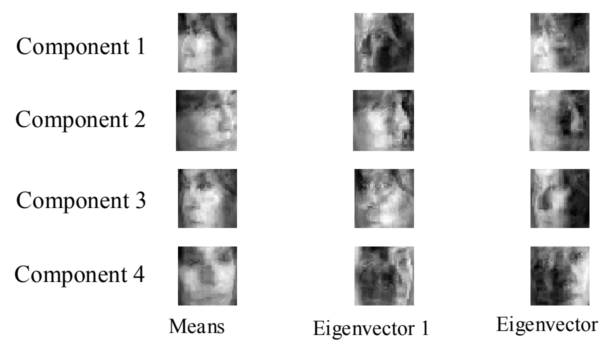

Figure 4. Sample means and eigenvectors

As can be seen from Figure 4, the means automatically converge to varying poses while the eigenvectors capture lighting variations among these poses. It is very interesting that this converged result agrees with semantic labeling of data into poses and illuminations.

We plug these models into the face recognition system shown in Figure 1 and evaluate the recognition results. We compare the performance of these models with the PCA with eight eigenvectors, making the same number of total eigenvectors for either model. These results are shown in Figure 5.

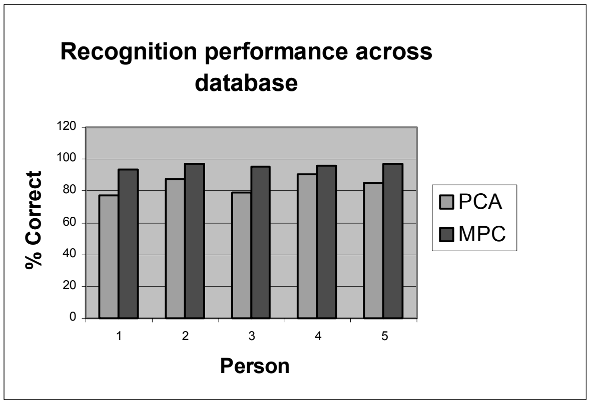

Figure 5. Face recognition performance

The overall recognition performance over the five subjects in the test set is 95.8% for the MPC while it is 83.8% for the PCA. The significant improvement in recognition performance is due to the better modeling performance of the MPC as opposed to the PCA. On average for this set of faces, the MPC has around 34% smaller representation error than the PCA, even with the same number of total eigenvectors. The confusion matrix for the test data is shown in Table 1. Each subject in the test has 143 images.

Table 1. Confusion matrix for face data

Subject	1	2	3	4	5
1	134	1	4	0	4
2	0	139	2	1	1
3	0	1	136	0	6
4	1	0	2	137	3
5	0	0	3	1	139

In Table 1, a row represents the number of times a subject is recognized as one of the other subjects. As may be seen, the diagonal entries are the largest values with some confusion due to the extreme lighting and pose variations.

4. CONCLUSION

We introduce a new statistical modeling technique, MPC that uses a mixture of eigenspaces to capture data variations efficiently. This model is a linear extension to the PCA and shows smaller representation error than the PCA for data with large variations in appearance. We illustrate the performance of this model on face data and show that the MPC has around 34% smaller reconstruction error for faces with varying poses and under different illumination conditions, even with the same number of total eigenvectors for both the models. The converged model parameters after training agree with semantic labeling of the data into poses and lighting variations. As shown in the paper, the means of the component eigenspaces converge to different poses in the data, while the eigenvectors for each eigenspace capture the lighting variations. We then plug this improved model into a face recognition system and test the performance on some faces from the PIE database. We show that the MPC has a recognition performance of 95.8%, as opposed to the PCA with only 83.8%.

MPC is a general statistical modeling tool and may be used to capture data variations in any generic data set. It is not restricted to faces or images. Some future directions of research include progressive training of the model and use of model for data compression.

5. ACKNOWLEDGEMENTS

The work done in this paper was supported in part by grants from General Motors.

6. REFERENCES

[1] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71-86, 1991.

[2] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. Wiley and Sons, New York, NY, 1973.

[3] T. Hastie and W. Stuetzle, "Principal curves," *Journal of the American Statistical Association*, vol. 84, pp. 502-16, 1989.

[4] E. Oja, "Neural networks, principal components and subspaces," *International Journal of Neural Systems*, vol. 1, pp. 61-68, 1989.

[5] S. Y. Kung and K. I. Diamantaras, "A neural network learning algorithm for adaptive principal component extraction (APEX)," Proc. of the IEEE ICASSP, pp. 861-64, 1990.

[6] T. Cox and M. Cox, *Multidimensional Scaling*, Chapman and Hall, London 1994.

[7] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, pp. 2323-26, December 2000.

[8] N. Kambhatla and T. K. Leen, "Dimension reduction by local principal component analysis," *Neural Computation*, vol. 9, pp. 1493-1516, 1997.

[9] M. Tipping and C. M. Bishop, "Mixtures of probabilistic principal component analyzers," *Neural Computation*, vol. 11, no. 2, pp. 443-82, 1999.

[10] D. Turaga and T. Chen, "Model-based error concealment for wireless video," submitted to *IEEE Trans. CSVT*, special issue on Wireless Video.

[11] T. Sim, S. Baker and M. Bsat, "The CMU pose, illumination and expression (PIE) database of human faces," Technical Report CMU-RI-TR-01-02, Robotics Institute, Carnegie Mellon University, January 2001.