Face Classification using a[1] Multiresolution Principal Component Analysis

Vic Brennan and Jose Principe
Computational NeuroEngineering Laboratory
Department of Electrical Engineering,
University of Florida, Gainesville, FL 32611
phone: 352-392-2662, fax: 352-392-0044
email: principe@cnel.ufl.edu

Abstract

Multiresolution Principal Component Analysis (M-PCA) uses Principal Component Analysis (PCA) to obtain multiresolution features for a signal. Bischof [1], [2] has used 3-layer networks to train Principal Component Pyramids for image compression. M-PCA uses a single computational layer adaptive linear network trained with the Generalized Hebbian Algorithm (GHA) [16]. The multiresolution features were applied to automatic face recognition and tested against the Olivetti Research Lab (ORL) database [8], [15], [18]. Classification with multiresolution had an average (over 10 runs) error rate of 2.4%.

Keywords

principal component analysis, multiresolution, automatic face recognition

Introduction

Reliable automated face recognition is useful in several applications [4, Table 1, p. 707]. One application is in access control systems. The ORL database has been used to test several face recognition algorithms (Table 1). The best results (error rate of 3.8%) used a convolutional network.All of the algorithms are considerably more computationally intensive than M-PCA. M-PCA is essentially Sanger's GHA network with partial connections to some output nodes. This paper will discuss multiresolution as a signal representation, review Sanger's GHA for PCA, present M-PCA as an extension of PCA, and discuss the design for using M-PCA features to classify the ORL database.

Multiresolution

A standard application of multiresolution is signal representation (e.g., an image) at different spatial scales. The ideas are demonstrated with a dyadic Haar decomposition (Figure 1)

The 1-D Haar basis vectors are,

$$e_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \frac{1}{\sqrt{2}}, \text{ and } e_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \frac{1}{\sqrt{2}}.$$

0-7803-5060-X/98/$10.00 © 1998 IEEE

Algorithm	Performance
Eigenfaces [18]	10%
Hidden Markov Model (HMM) [15]	5.5%
Self-Organizing Map + Convolutional Net [8]	5.75% (3.8%)
M-PCA	2.4% (0 - 7%)

TABLE 1
ERROR PERFORMANCE OF SEVERAL FACE RECOGNITION ALGORITHMS

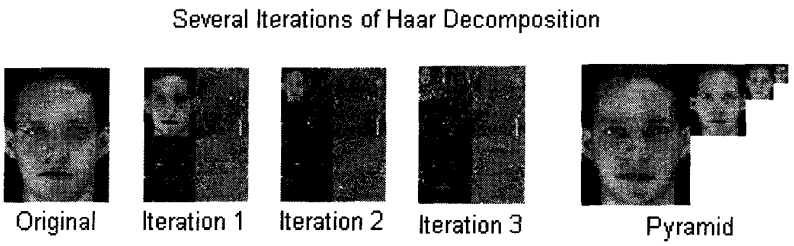


Fig. 1. Three Levels of Decomposition on the Approximation

For two dimensions, the (separable) basis vectors are,

$$w_1 = e_1 e_1^{'} \quad w_2 = e_1 e_2^{'} \quad w_3 = e_2 e_1^{'} \quad w_4 = e_2 e_2^{'}$$

Compression is done iteratively; each iteration produces a higher level of compression. At the first iteration, the input is the level 0 (original) image (denoted L0 for convenience). The input is parsed into non-overlapping 2×2 blocks. Each block is projected against w_1, w_2, w_3, and w_4. Each projection against a basis function results in a compressed image of half the length and half the width of the input image. For the Haar basis, the projections against w_1 form the best approximation at level 1 (L1). The other projections are called detail signals. The approximation and the detail signals needed to reconstruct the original image don't take any more pixels (memory or BW) than the original image. Subsequent iterations use the approximation signal as an input to get the next level of approximation and details. All the information for creating the first (level 1) approximation is contained in the original (level 0) image. Similarly, an approximation at any level only uses information from the approximation at the previous level. Lower level approximations have more spatial resolution and less compression than high level approximations. Clearly, there is less data to process if classification can be performed with one of the compressed (higher level) subimages.

In general, fixed multiresolution decompositions can be done by taking iterations of a transform. The choice of the basis function affects image

quality (e.g., artifacts such as blurring, blocking, ...). Unitary transforms decorrelate components and are easy to invert (reconstruct the input image from the approximation and detail signals). However, a disadvantage of using fixed bases is that they are independent of the input signal and cannot take advantage of input statistics.

PCA Networks

PCA can be used for deriving adaptive basis functions (the basis functions themselves carry information about the signal). A PCA (Karhunen-Loeve) decomposition uses a unitary transform and is optimal for selecting high energy components. Standard GHA uses a fully connected architecture to retrieve the (ordered) eigenvectors of a signal's autocorrelation matrix.

$$R = W\Lambda W^T = \sum_k \lambda_k w_k w_k^T \tag{1}$$

The k×k autocorrelation matrix R is Toeplitz, and each of the matrices in the summation is doubly symmetric (but not generally Toeplitz). The weights are updated with,

$$y_j(n) = \sum_{i=0}^{p-1} w_{ji}(n)x_i(n), \text{calculate output} \tag{2}$$

$$\Delta w_{ji}(n) = \eta y_j(n)\left[x_i(n) - \sum_{k=0}^{j} w_{ki}(n)y_k(n)\right], \text{update weights} \tag{3}$$

PCA is an optimal feature extractor for signal representation, but is sub-optimal for classification [7]. On the other hand, extensive literature [10], [17] shows that multiresolution representations provide good features for classification. Each multiresolution subimage represents information about the image at different spatial resolution and spatial frequency. Images from a given class may resemble those of another class at one resolution and those of yet another class at a different resolution. An image should resemble other images from its only own class across all resolutions. By combining classifications performed on different projections, the misclassifications should whiten out and the correct classifications reinforce.The goal of this paper is to combine the strengths of both methods to produce a feature extraction stage that creates maximum energy multiresolution features.

M-PCA Decomposition

M-PCA utilizes the straight GHA for weight update. The only difference is that the network topology is only partially connected to produce a multiresolution representation of the input signal. Figure 2 shows the partial connectivity. While each output node of the PCA network is connected to all the inputs, in M-PCA only a few nodes are connected to all inputs. The missing connections can be interpreted as zero weights in a fully connected

architecture. Since GHA adapts the weights independently of their values, GHA can be applied without modification to M-PCA. With wavelets,

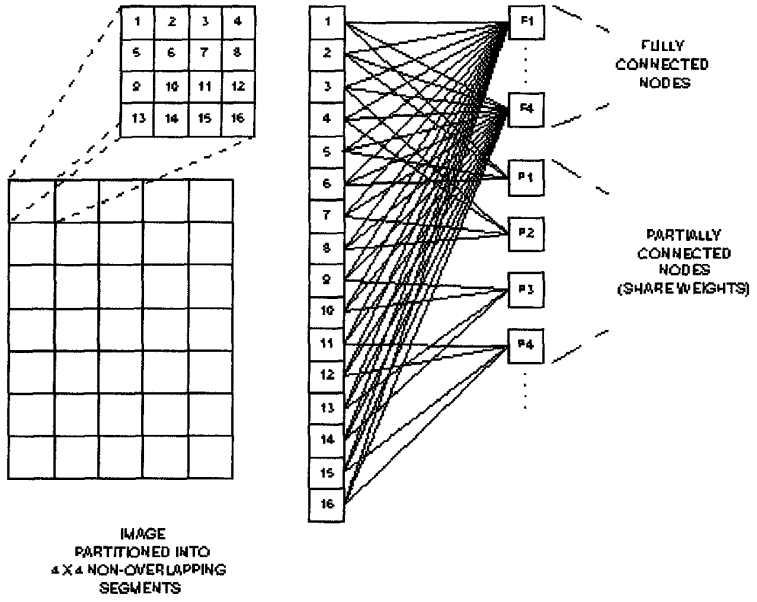


Fig. 2. M-PCA Network

the shorter basis functions are translated to span the input (resulting in more frequent updates for high frequency components). For comparison to standard multiresolution bases, we chose to emulate the translation of short basis functions by constraining GHA to share weight between some partially connected nodes. The multiresolution decomposition of the autocorrelation matrix is only slightly more complicated to analyze. For a time (1-D signal) signal, three steps are required to halve the resolution,

1. Deflate the matrix - the resulting matrix $R^{'}$ is doubly symmetric and the underline denotes that the matrix has been flipped left-to-right and up-to-down.

$$R^{'} = R - \sum_{\alpha} \lambda_{\alpha} w_{\alpha} w_{\alpha}^{T} = \begin{bmatrix} A & B \\ \underline{B} & \underline{A} \end{bmatrix} \tag{4}$$

2. Project to a lower dimension - the matrix $R^{''}$ is also doubly symmetric and of half the order of R.

$$R^{''} = A + \underline{A} \tag{5}$$

3. Find the (lower resolution) eigenvectors of R''. If desirable, sum along the diagonals of R'' to find Toeplitz autocorrelation matrix R''' of the deflated signal.

For images (2-D signals) the process is done twice (halve both the x and y dimensions). The calculations for 2-D signals are further complicated by the geometry of the segmentation (e.g., there are several ways to segment a 16×16 block of data into four subsegments).

M-PCA decomposition does not result in an orthogonal basis but in a normalized spanning set. While the features associated with using a unitary transform are lost, the impact is minimal since reconstruction (synthesis) is not required for classification. The trade-off for the loss of orthogonality is that components are ordered by energy across several resolutions.

M-PCA Decomposition for the ORL Database

The Olivetti Research Lab (ORL) has a face database containing 10 different pictures of 40 people. The images are 112×92, 8-bit grayscale. The pictures show variation in background lighting, scale, orientation, and facial expression (figure 3). Individuals who used eyeglasses were allowed to pose both with and without eyeglasses. Some individuals looked very similar to others. Each image was cropped to 112×80 and segmented into

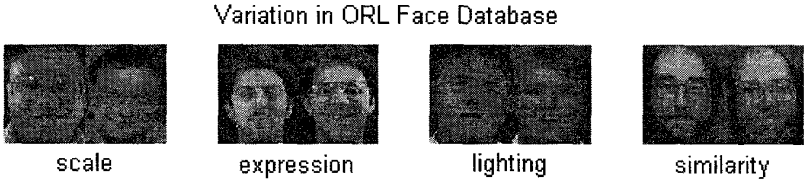


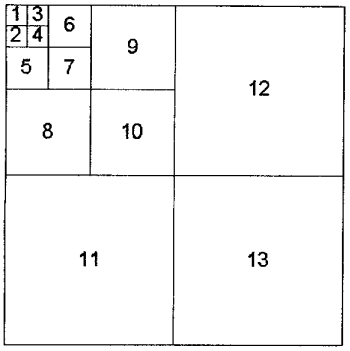
Fig. 3. Varying Conditions in ORL Pictures

non-overlapping 16x16 blocks. The resolutions of the components were four at 16×16, three at 8×8, three at 4×4, and three at 2×2 (figure 4). Each block was raster scanned left-to-right and top-to-bottom. The choice and ordering of the resolutions was just to mimic the Haar decomposition for later comparisons. Figure 5 shows a sample decomposition along with the original image.

M-PCA Classification for the ORL Database

The goal of this experiment was to find a simple classifier, then compare the performance of the classifier using single resolution features against the performance using multiresolution features. Our system uses a single computational layer linear network for feature extraction, and several independent linear networks for classification.

The 112×80 pixel images are decomposed into four 7×5, three 14×10, three 28×20, and three 56×40 images (Figure 4). The overall strategy is to



1, 2, 3, 4 are level 4 projections along 16x16 eigenvectors, and have 7x5 pixels.

5, 6, 7 are level 3 projections along 8x8 eigenvectors, and have 14x10 pixels.

8, 9, 10 are level 2 projections along 4x4 eigenvectors, and have 28x20 pixels.

11, 12, 13 are level 1 projections along 2x2 eigenvectors, and have 56x40 pixels.

Components are numbered by decreasing energy

Fig. 4. Selected Resolutions

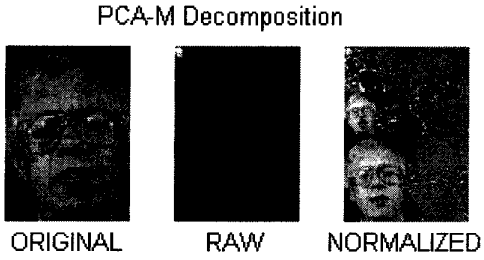


Fig. 5. M-PCA Decomposition of One Picture

extract a given number of eigenvectors, deflate the space, then project to a lower dimension space. A set of partially connected nodes share weights to span the full input. There is an inherent trade-off in spatial resolution and frequency resolution. Components 1 to 4 have the longest eigenvectors (highest frequency resolution) and the lowest spatial resolution. Components 11 to 13 have the shortest eigenvectors (lowest frequency resolution) and the highest spatial resolution. By design components are ordered by decreasing energy. However, the highest energy component at scale L3 may be higher than the lowest energy component at scale L4.

A linear template-matching classifier was used for each feature. That is, since the image was split into thirteen multiresolution subimages (4), thirteen classifiers attempted to identify the person from each of the respective subimages. Each classifier has 40 output nodes (one for each class/person), the number of input nodes depends on the resolution of the input feature. Let x represent a feature (subimage) of the input image, then the equations

for the output nodes of an individual classifier are,

$$y_j = w_j^T x. \tag{6}$$

The weights for each node are obtained by averaging the appropriate subimages from the training set. The classifier chooses the class with the highest output.

$$\text{Class } k : y_k = \max_{\forall j \in K} \{y_j\}. \tag{7}$$

An additional majority vote layer combines the results of the thirteen feature classifiers. This use of multiresolution features is much less computationally intensive than the other approaches of Table 1. The overall structure can also be presented as a 2-layer hierarchical One-Class-One-Network (OCON) Decision-Based Neural Network (DBNN) [11, pp. 118-120].

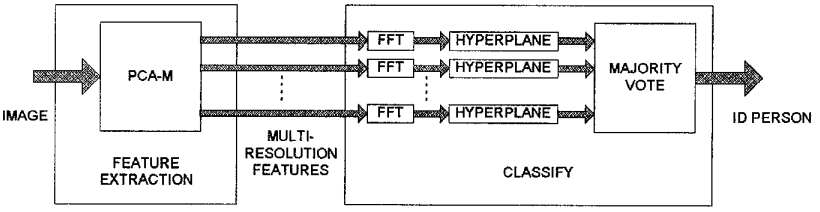


Fig. 6. Classification with M-PCA

Results

The original scheme was to use a bank of eigenfilters each tuned to a different class of images. The eigenvectors were similar, often differing only in order. We chose to use the eigenvectors from a single image and base classifications on the projections against the single set of eigenvectors. When a misclassification occurred, we observed low variance across the output nodes, and that the second best guess was often correct.

As a baseline, the classifier was tested against the raw image resulting in a 95% error. The classifier was then used with raw (no FFT) multiresolution features resulting in a 20% error rate. A concern was that the feature classifiers might be redundant, that is, have the same correct and incorrect classifications. Fortunately, there was not a large overlap in classification. Using different multiresolution features added new information. At this point, three issues were explored,

1. Optimally combining the information from the different experts: A Probably Approximately Correct (PAC) Boosting Algorithm [6] is an algorithm which can be used to optimally weight the ensemble of experts. Unfortunately with a multiresolution scheme, the performance in the training

data does not consistently reflect generalized performance (memorization increased with spatial resolution). No better alternative than a simple majority vote was found (or needed).
2. Improving the performance of the feature classifiers: Although there are probably better ways to improve the performance of the feature classifiers, it seemed intuitive to obtain some shift invariance by representing the data in autocorrelation or frequency space. Both autocorrelation and magnitude of the FFT gave similar performances. Magnitude of the FFT was used since symmetries allowed a further halving of nodes. Taking the magnitude of the FFT of the multiresolution features dropped the errors to 7%.
3. Determining the sensitivity to selection of the training set: Initially, the first 5 images of each person were used for training. After examining the errors, it seemed that performance might be sensitive to the partitioning of data to test and training sets. Several runs with random selections of training data were tried. Table 2 shows typical results over five runs. There were 200 images in each of the training and test sets. The first column shows results of using just the highest energy component at each of the 4 resolutions. The second column shows results with all components. Table 3 shows results averaged over 20 different partitions.

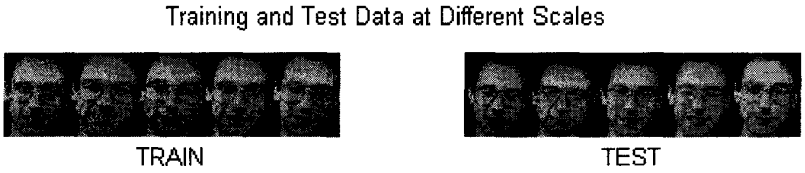


Fig. 7. Training and Test Data at Different Scales

Permutation	High 4	All	Rate
none	0/22	0/14	7.0%
1	0/8	0/5	2.5%
2	0/11	0/4	2.0%
3	0/7	0/2	1.0%
4	0/7	0/2	1.0%

TABLE 2
ERRORS WITH RANDOM PERMUTATIONS OF TRAINING SET IMAGES

As we should have anticipated, the small size of the data set made classification sensitive to the selection of the training set. For some classes, it was possible to select a partitioning such that the training and test sets would be obviously different in scale or lighting (Figure 7). We observed that some permutations of training sets had no errors. The existence of

Features	High Energy Train/Test (Error)	All Train/Test (Error)
M-PCA, Spatial	2 / 49 (24.5%)	3 / 44 (22.0%)
M-PCA, abs(FFT)	0 / 6 (3.0%)	0 / 4.8 (2.4%)
Haar Decomposition	0 / 13 (5.5%)	0 / 5 (2.5%)

TABLE 3
PERFORMANCE SUMMARY (AVERAGED OVER 20 TEST/TRAIN PARTITIONS)

such permutations showed that the output space was linearly separable. Further, the low average error rate (over 20 random partitions) indicated that it was possible to find the "good" weights most of the time.

We also tried to increase the size of the training set by using deformations (reflections) of the original training data [5]. However, no improvement was observed.

CONCLUSION

Since the HMM and SOFM with convolutional network got $\sim 96\%$ correct classification, we thought that the ORL database would be good test case for M-PCA. We were hoping to see an improvement from using multiresolution over single resolution, but the low error rates were unexpected. We expected to implement M-PCA with overlapping inputs, some image segmentation, stronger classifiers (e.g., MLP), and better weighting for the ensemble of classifiers.

It seems that the good performance came from using the FFT, using multiple features, and using multiresolution features. Combining the outputs of several classifiers works best when the classifiers are well differentiated (training sets or algorithms). Each M-PCA feature is differentiated by frequency and resolution. M-PCA should perform better than fixed multiresolution bases when the classes have different second-order statistics (different eigenvectors). Using M-PCA also makes the classifier less dependent (compared to single resolution PCA) on the choice of resolution. Although an energy is not an optimal feature for selecting components for image quality, it seems to be useful for classification. At each resolution, the classifier with the lowest error rate used the feature with the highest energy (variance).

M-PCA performed favorably when compared to fixed-basis multiresolution and single resolution PCA. A final advantage of M-PCA was that since a template-matching linear-basis classifier was used, the update procedure is straightforward. It is easy to add new people into the classifier or accommodate long-term changes in a persons appearance.

References

[1] Horst Bischof, *Pyramidal Neural Networks*, 1995, Lawrence Erlbaum Associates, Mahwah, NJ.

[2] Horst Bischof and Kurt Hornik, "PCA-Pyramids for Image Compression", *Advances in Neural Information Processing Systems*, 7:941-948, 1996.

[3] R. Brunelli and T. Poggio, "Face Recognition Features versus Templates", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(10):1042-1052, October 1993.

[4] Rama Chellappa, Charles Wilson, and Saad Sirohey, "Human and Machine Recognition of Faces: A Survey", *Proceedings of the IEEE*, 83(5), May 1995.

[5] Harris Drucker, Robert Schapire, and Patrice Simard, "Boosting Performance in Neural Networks", *Advances in Pattern Recognition Systems using Neural Network Technologies*, 1993, World Scientific Publishing, Singapore.

[6] Yoav Freund and Robert Schapire, "A Decision-Theoretic Generalization of On-line Learning and an Application to Boosting", *Proceedings of the Second European conference on Computational Learning Theory*, March 1995.

[7] Keisnosuke Fukunaga, *Introduction to Statistical Pattern Recognition*, 1990, Academic Press, New York, NY.

[8] C. Lee Giles, Steve Lawrence, and Ah Chung Tsoi, "Convolutional Networks for Face Recognition", *Proceedings of the Ninth Yale Workshop on Adaptive and Learning Systems*.

[9] Simon Haykin, *Neural Networks, A Comprehensive Foundation*, 1994, McMillan Publishing Company, Englewood Cliffs, NJ

[10] Anil K. Jain, *Fundamentals of Digital Image Processing*, 1989, Prentice-Hall, Englewood Cliffs, NJ.

[11] S. Y. Kung, *Digital Neural Networks*, 1993, PTR Prentice Hall, Englewood Cliffs, NJ.

[12] John Makhoul, "On the Eigenvectors of Symmetric Toeplitz Matrices", *IEEE Transactions on Acoustices, Speech, and Signal Processing*, vol assp-36(4):868-876 August 1981.

[13] Francesco Palmieri, "Anti-Hebbian Learning in Topologically Constrained Linear Networks: A Tutorial", *IEEE Transactions on Neural Networks*, 4(5):748-761, September 1993.

[14] Francesco Palmieri, "Self-Association and Hebbian Learning in Linear Neural Networks", *IEEE Transactions on Neural Networks*, 6(5):1165-1184, September 1995.

[15] F.S. Samaria, *Face Recognition using Hidden Markov Models*, PhD thesis, Trinity College, University of Cambridge, 1994.

[16] T. Sanger, "Optimal Unsupervised Learning in a Single Layer Feedforward Network", *Neural Networks* 12:459-473, 1989.

[17] Gilbert Strang and Truong Nguyen, *Wavelets and Filter Banks*, 1996, Wellesley-Cambridge Press, Wellesley, MA.

[18] M. Turk and A. Pentland, "Eigenfaces for Recognition", *Journal of Cognitive Neuroscience*, 3:71-86, 1991.

[19] Martin Vetterli and Jelena Kovacevic, *Wavelets and Sub-band Coding*, 1995, Prentice-Hall, Inc., Englewood Cliffs, NJ.