System Level Design of Real Time Face Recognition Architecture Based on Composite PCA

Rajkiran Gottumukkal, Vijayan K. Asari

Department of Electrical and Computer Engineering, Old Dominion University

{rgott002,vasari}@odu.edu

ABSTRACT

Design and implementation of a fast parallel architecture based on an improved principal component analysis (PCA) method called Composite PCA suitable for real-time face recognition is presented in this paper. The proposed architecture performs the tasks of both feature extraction and classification. Composite PCA takes in to consideration the local features of face images, which do not vary widely between face images of the same person taken under varying expression, illumination and pose. Hence it leads to a better recognition rate than PCA. Composite PCA has more parallelism than conventional PCA and this parallelism is utilized to design an efficient architecture capable of performing real-time face recognition. The face recognition system is implemented in an FPGA environment and tested using standard databases. The system is able to recognize a person from a database of 110 images of 10 individuals in approximately 4 ms.

Categories and Subject Descriptors

C.3 [**Special-Purpose and Application-Based Systems**]: Signal processing systems

General Terms

Algorithms, Design, Security

Keywords

Face recognition, Principal Component Analysis, Parallel computer architectures

1. INTRODUCTION

Automatic identification of the human faces has been an area of active research for the past few years. Although recognition of faces does not seem to present any difficulties for most human observers, computerized face recognition systems cannot achieve satisfactory performance due to the generally similar shape of the faces combined with the numerous variations in expression, illumination and pose between images of the same face. This task, therefore, still presents a significant challenge and is considered as one of the fundamental problems in pattern analysis. In addition to the importance of the face recognition task from a research point of view, it has a number of commercial and law enforcement applications such as entrance control in buildings, access control for computers in general or for automatic teller machines in particular, day-to-day affairs like withdrawing money from a bank account or dealing with the post office, or in the prominent field of criminal investigation. Even when acceptable recognition has been accomplished in simulation on computers, the actual implementation typically requires long run times on high performance workstations or the use of expensive supercomputers. The goal of this work is to develop an efficient face recognition system that would be able to recognize a person in less than one thirtieth of a second so that we can process every frame from a surveillance camera or a similar device.

There are two main approaches for face recognition [1-2]. The first approach is the feature-based matching approach using the relationship between facial features such as eyes, mouth and nose [2-3]. The second approach is the template matching approach using the holistic features of face image [2][4-7]. Template-based techniques often follow the subspace method called eigenface originated by Turk and Pentland [4]. This technique is based on the Karhunen-Loève transformation, which is also referred to as PCA, and was introduced into face processing by Kirby and Sirovich [6]. It has gained great success and become a de facto standard and a common performance benchmark in face recognition. The work in this paper is based on the template-based recognition scheme called PCA.

2. COMPOSITE PCA

The PCA based face recognition method is not very effective under varying expression, illumination and pose since it considers the global information of each face image and represents them with a set of weights. Under the conditions of varying expression, illumination or pose these weight vectors will vary considerably form the weight vectors of the images with normal expression, illumination and pose, hence it is difficult to identify them correctly. On the other hand if the face images are divided into smaller regions and the weight vector are computed for each of these regions, then the weights will be more representative of the local information of the face. When there is a variation in the expression, illumination or pose, only some of the face regions will vary and rest of the regions will remain the same as the face regions of a normal image. Hence weights of the face regions not affected by these variations will closely match with the weights of the same individuals face regions under normal conditions. Therefore it is expected that better positive recognition rates can be obtained by following this approach and we refer to this approach as Composite PCA. We expect that if the face images are divided into very small regions the global information of the

GLSVLSI'03, April 28-29, 2003, Washington, DC, USA.

face will be completely lost and the positive recognition rates may deteriorate. The remainder of this section elaborates on the Composite PCA method.

In this method each image is divided into N^2 smaller regions and as before the original image is considered to be a matrix of size $L \times L$. Hence the size of each sub-image will be L^2 / N^2. These sub-images can be represented as a function of the original image, I_i as shown below:

$$I_{ijk}(m,n) = I_i\left(\frac{L}{N}(j-1)+m, \frac{L}{N}(k-1)+n\right) \qquad \forall i,j,k \tag{1}$$

where i varies from 1 to M, M being the number of images in the training set, j and k vary from 1 to N, N^2 being the number of sub-images and m and n vary from 1 to L/N. The average image of all the test sub-images is computed as below:

$$A = \frac{1}{M \cdot N^2} \sum_{i=1}^{M} \sum_{j=1}^{N} \sum_{k=1}^{N} I_{ijk} \tag{2}$$

The next step is to normalize each training sub-image by subtracting it from the average as shown below:

$$Y_{ijk} = I_{ijk} - A \qquad \forall i,j,k \tag{3}$$

From the normalized images the covariance matrix is computed as shown:

$$C = \frac{1}{M \cdot N^2} \sum_{i=1}^{M} \sum_{j=1}^{N} \sum_{k=1}^{N} Y_{ijk} \cdot Y_{ijk}^{T} \tag{4}$$

C will be a matrix of size $L^2/N^2 \times L^2/N^2$; hence the size of the covariance matrix is reduced by a factor of N^4 compared to the covariance matrix computed for the original PCA method. Next the eigenvalues and eigenvectors of C are computed, and M' eigenvectors corresponding to the M' largest eigenvalues are considered to compute the weights of the training sub-images as shown:

$$W_{pnjkr} = E_r^{T} \cdot \left(I_{pnjk} - A\right) \qquad \forall p,n,j,k,r \tag{5}$$

where r varies from 1 to M', n varies from 1 to Γ, Γ being the number of images per individuals, and p varies from 1 to P, P being the number of individuals in the training set. The test image is also divided in to N^2 sub-images and the weights are computed for each of the sub-image using the eigenvectors as shown in the next equation:

$$Wtest_{jkr} = E_r^{T} \cdot \left(Itest_{jk} - A\right) \qquad \forall j,k,r \tag{6}$$

The mean of the weights of all the sub-images belonging to an individual are computed as shown below:

$$T_{pjkr} = \frac{1}{\Gamma} \sum_{n=1}^{\Gamma} W_{pnjkr} \qquad \forall p,j,k,r \tag{7}$$

The minimum distance is computed as show below:

$$D_{pjk} = \frac{1}{M'} \sum_{r=1}^{M'} \left| Wtest_{jkr} - T_{pjkr} \right| \qquad \forall p,j,k \tag{8}$$

$$D_p = \frac{1}{N^2} \sum_{j=1}^{N} \sum_{k=1}^{N} D_{pjk} \qquad \forall p \tag{9}$$

The minimum value of D_p corresponds to the p^{th} individual in the training set whose face resembles closest to the face in the test image. The minimum distance is compared with a pre-computed threshold value, which is obtained by performing a series of training stages on various face images. If it exceeds the threshold value the test image is rejected. On the other hand, if it is less than or equal to the threshold it is considered as a recognition.

2.1 Conventional PCA vs. Composite PCA

We compared the accuracy of conventional PCA method and Composite PCA method by testing them on three face databases. All images in the databases are matrices of size 64×64. Only the first 20 eigenvectors were considered for the testes, i.e. $M'=20$. It was observed that the recognition rate deteriorates for $N^2 \geq 256$ since some of the local information of the face is lost when the face images are divided into very small sub-images. Hence in this paper we only analyze the results obtained for $N^2<256$.

The ODU database is used to test the methods under varying facial expressions. It has face images of 20 individuals, each individual having 10 images in the database. Each of the face images of an individual in the database has some variation in the facial expression. Of the 10 images of each person in the face database, 8 images were used to train the algorithm and all the images were used for testing. Table 1 shows the results obtained.

Table 1. Performance of Composite PCA with varying N^2 on ODU database

	PCA	Composite PCA		
		N^2=4	N^2=16	N^2=64
Positive recognition rate	98.13	99.38	100	100
Negative recognition rate	1.88	0.63	0.0	0.0
False rejection rate	0.0	0.0	0.0	0.0

The second database used for comparing the two methods is called the Yale database and was obtained from [8]. It has images of 15 individuals, each individual having 11 images. The face images vary with respect to face expression and illumination. Of the 11 images of a person, 8 were used for training and to test the recognition rates all the images were used. The performance of the two methods with Yale database is shown in Table 2.

Table 2. Performance of Composite PCA with varying N^2 on Yale database

	PCA	Composite PCA		
		N^2=4	N^2=16	N^2=64
Positive recognition rate	84.24	87.88	97.58	96.36
Negative recognition rate	12.12	3.64	1.82	2.42
False rejection rate	3.64	8.48	0.61	1.21

The third database used for comparing the two methods is called the UMIST database and was obtained from [9]. This database has 120 images of 20 individuals, each individual having 6 images taken at a different pose with a normal expression. Out of the six images of a person, only four were used for training and to test the recognition rates all the images were used. The performance of the two methods with UMIST database is shown in Table 3.

Table 3. Performance of Composite PCA with varying N^2 on UMIST database

	PCA	Composite PCA		
		N^2=4	N^2=16	N^2=64
Positive recognition rate	81.67	86.67	81.67	90.0
Negative recognition rate	18.33	11.67	13.33	5.0
False rejection rate	3.64	1.67	5.0	5.0

The most important parameters in face recognition are positive recognition rate, negative recognition rate and false rejection rate in that order. By using the Modular PCA method we can achieve an improvement in these parameters at N^2=16 for ODU and Yale database and at N^2=64 for UMIST database compared to the conventional PCA method. Hence the proposed method provides an overall improvement over the PCA method and the optimum value of N^2 can be fixed at 64.

3. DESIGN OF A PARALLEL ARCHITE-CTURE BASED ON COMPOSITE PCA METHOD

3.1 Modified Distance Equation

By performing simple rearrangement of the terms in the equations to be implemented the design was made robust by avoiding some complex operations. The equation to compute the distance of the weight vectors of test image and images in database is simplified as shown below, and then the design of the parallel architecture is performed.

For a test image the architecture should be able to compute the weight vector $Wtest_{jkr}$ defined in equation (6) and compute the distance between the weights of the test image and the mean weights of the individuals in the database using equation (8). Substituting $Wtest_{jkr}$ from equation (6) in equation (8) gives:

$$D_{pjk} = \frac{1}{M'} \sum_{r=1}^{M'} \left| E_r^{\,T} \cdot \left(Itest_{jk} - A \right) - T_{pjkr} \right| \qquad \forall p, j, k \quad (10)$$

This can be rearranged as shown below to group the terms than can be computed off-line:

$$D_{pjk} = \frac{1}{M'} \sum_{r=1}^{M'} \left| E_r^{\,T} \cdot Itest_{jk} + \left(-E_r^{\,T} \cdot A - T_{pjkr} \right) \right| \quad \forall p, j, k \quad (11)$$

The term $\left(-E_r^{\,T} \cdot A - T_{pjkr} \right)$ can be computed off-line since all the values are available off-line. Hence the term can be expressed as a constant as shown below:

$$C_{pjk} = \left(-E_r^{\,T} \cdot A - T_{pjkr} \right) \qquad \forall p, j, r \qquad (12)$$

Then the equation (11) can be rewritten as:

$$D_{pjk} = \frac{1}{M'} \sum_{r=1}^{M'} \left| E_r^{\,T} \cdot Itest_{jk} + C_{pjk} \right| \qquad \forall p, j, k \qquad (13)$$

In equation (13) the division by M' can be avoided for the sake of reducing the hardware, and since it acts only as a scaling factor the accuracy of the system will not be compromised. Hence the final equation to be implemented is:

$$D_{pjk} = \sum_{r=1}^{M'} \left| E_r^{\,T} \cdot Itest_{jk} + C_{pjk} \right| \qquad \forall p, j, k \qquad (14)$$

A parallel architecture is designed to implement the equation (14), which gives the distance of the weight of test image from weights of each of the individuals in the database.

3.2 Parallel architecture based on Composite PCA

The block diagram of the parallel architecture to compute the distance using Modular PCA method is shown in Figure 1. Two buffers are used to store two consecutive eigenvectors $E_r^{\,T}$ and $E_{r+1}^{\,T}$. But at any give time only one eigenvector will be used to compute the distances. The advantage of using two buffers is that, while one is being used to compute the distances, the other can be loaded with the next eigenvector form the ROM. Hence the time to load the buffers will not affect the speed of computation of the distances. The test image is divided into sub-images and loaded into separate buffers. The detailed block diagram of the programming elements PE1 and PE2 are shown in Figure 2.

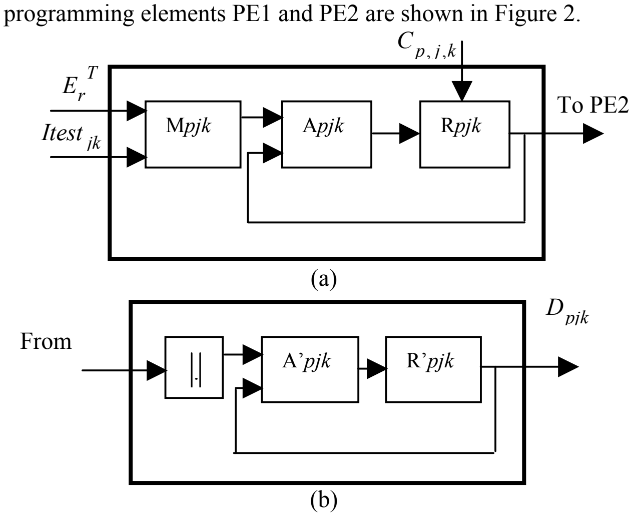

Figure 2. Block diagram of (a) PE1, (b) PE2.

The constant term C_{pjk}, defined in equation (12) is loaded into a register set. These constants are loaded into the corresponding registers R*pjk*, which are used to accumulate the value of ($E_r^{\,T} \cdot Itest_{jk} + C_{pjk}$). The absolute value of R*pkj* is accumulated in the register R'*pjk*, which is a distance of the first image weight and the weights in the database. After these sequences of events are completed, a signal is sent to the multiplexer, which will select the next eigenvector and the process is repeated. When all the eigenvectors are used, the final distances will be available at the outputs of R'*pjk* registers. The comparator finds the minimum distance and compares it with a fixed threshold. If the minimum distance is less than or equal to the threshold, the individual corresponding to that distance is recognized as the test face. If the minimum distance is greater than the threshold then the test face is rejected.

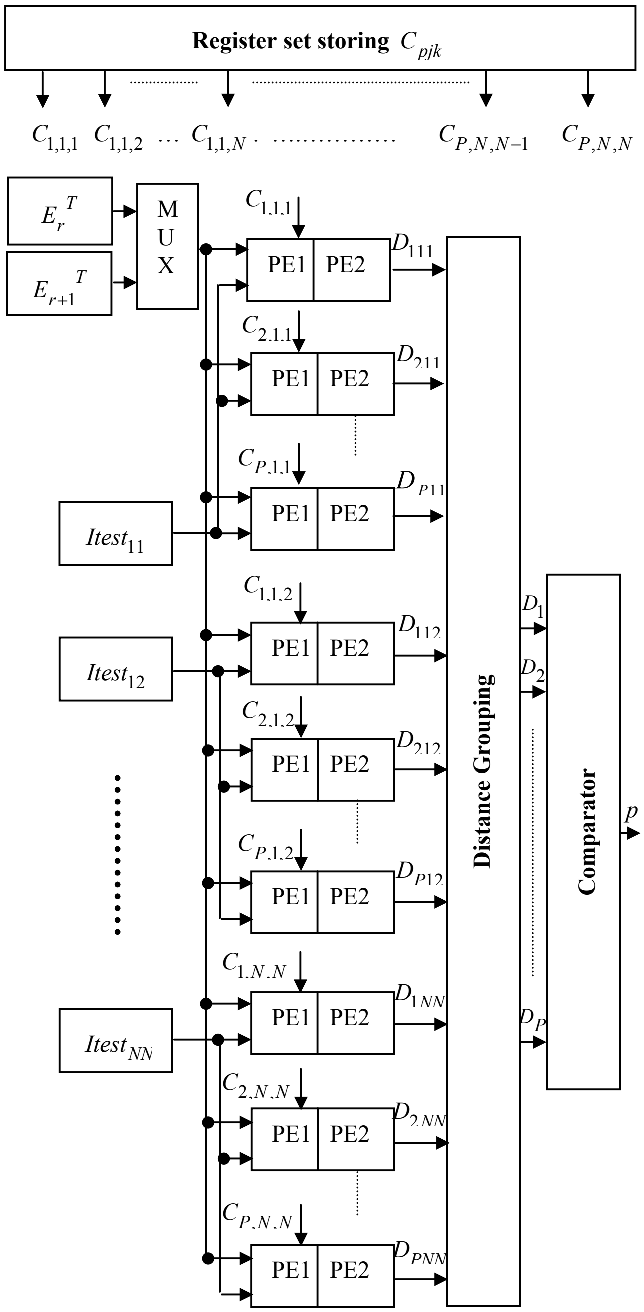

Figure 1. Face recognition architecture based on Composite PCA.

The parallel architecture shown in Figure 1 was simulated using Altera's Quartus II, for P=10 and N=16. The details of the number of logic gates and flip-flops in the FPGA implementation of the proposed architecture are given in Table 4.

Table 4. FPGA implementation results

Device name	EP20K100ETC144-1X
Total logic elements	7820
Flip-flops	2348
Clock frequency	100MHz
Recognition time	3880us

4. CONCLUSION

The concept of PCA was modified to arrive at a new method for feature extraction called Composite PCA. We showed that the new method is more accurate than the conventional PCA method and has more inherent parallelism. We modified the distance equation to arrive at a simplified equation, which has simplified the design of hardware to implement it. High-speed parallel hardware architecture to perform face recognition based on Composite PCA has been presented. The architecture was able to perform face recognition on a database consisting of 10 individuals in 3880us. This face recognition architecture is very well suited for use in real-time face recognition.

5. REFERENCES

[1] R. Chellappa, C.L. Wilson, and S. Sirohey. Human and machine recognition of faces: a survey. Proc. of IEEE, vol. 83, No. 5, pp 705–740, May 1995.

[2] R. Brunelli and T. Poggio. Face recognition: feature versus templates. IEEE Trans. Pattern Anal. Machine Intell. 15 (10), 1042–1052, 1993.

[3] L. Wiskott, J.M. Fellous, N. Kruger and C. Malsburg. Face recognition by elastic bunch graph matching. IEEE Trans. Pattern Anal. Machine Intell, 19 (7), 775–780, 1997.

[4] M. Turk and A. Pentland. Face recognition using eigenfaces. Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 586–591, 1991.

[5] P. Belhumeur, J. Hespanha and D. Kriegman. Eigenfaces vs. fisherfaces: class specific linear projection. IEEE Trans. Pattern Anal. Machine Intell. 19 (7), 711–720, 1997.

[6] M. Kirby and L. Sirovich. Application of the Karhunen–Loeve procedure for the characterization of human faces. IEEE Transactions on Pattern Analysis and Machine Intelligence 12 (1), 103–108, 1990.

[7] D.B. Graham and N.M. Allinson. Characterizing Virtual Eigensignatures for General Purpose Face Recognition. Face Recognition: From Theory to Applications, NATO ASI Series F, Computer and Systems Sciences, Vol. 163.

[8] http://cvc.yale.edu/projects/yalefaces/yalefaces.html

[9] http://images.ee.umist.ac.uk/danny/database.html