

A Flexible and Efficient Hardware Architecture for Real-Time Face Recognition Based on Eigenface

Hau T. Ngo, Rajkiran Gottumukkal and Vijayan K. Asari
Department of Electrical and Computer Engineering
Old Dominion University, Norfolk, VA 23529
{hngox001, rgott002, vasari}@odu.edu

Abstract

We describe a flexible and efficient multilane architecture for real-time face recognition system based on modular Principal Component Analysis (PCA) method in a Field Programmable Gate Array (FPGA) environment. We have shown in [1] that modular PCA improves the accuracy of face recognition when the face images have varying expression and illumination. The flexible and parallel architecture design consists of multiple processing elements to operate on predefined regions of a face image. Each processing element is also parallelized with multiple pipelined paths/lanes to simultaneously compute weight vectors of the non-overlapping region, hence called multilane architecture. The architecture is able to recognize a face image from a database of 1000 face images in 11ms.

1. Introduction

Automatic recognition of faces is considered as one of the fundamental problems in computer vision and pattern analysis, and many scientists from different areas have addressed it [2-3]. One of the methods that yields promising results on frontal face recognition is the PCA method, which is a statistical approach where face images are expressed as a subset of their eigenvectors, and hence called eigenfaces [4]. In this paper, we use a modular version of the PCA method; known as modular PCA to perform face recognition. It is shown to be more accurate than the PCA method for faces with varying expression and illumination [1]. A hardware design of a multilane architecture for modular PCA is presented in this paper. The architecture is capable of performing face recognition in real time even for large databases. We implemented the hardware on an FPGA due to its flexibility and cost considerations [5-6].

2. Architecture for Modular PCA

In modular PCA, each face image in the training database is divided into N^2 sub-images and these sub-images are represented by a set of weight vectors which are computed based on eigenvectors and eigenvalues. Similarly, in the testing phase, the test image I_{test} is divided into N^2 sub-images and the weight vector W_{test} is computed for each sub-image. The minimum distance between the test image's weight vector and the weight vectors of the images in the database is determined and it is compared with a pre-computed threshold value for classification. The training phase needs to be done only once for a particular training database, hence it can be done offline and it does not affect the face recognition time. Obviously the testing phase needs to be done each time a test image needs to be recognized.

It was shown in [5] that Euclidian distance between the test image's weight vector and the weight vectors of the images in the database can be computed by

$$D_{ijk} = \sum_{r=1}^{M'} \left| E_r^T \cdot I_{test\ jk} + C_{ijk} \right| \quad \forall i, j, k \quad (1)$$

where E is the eigenvector, j and k are the indices of the sub-image, M' is the number of eigenvectors used to compute the weight vector W and C_{ijk} is defined as

$$C_{ijk} = \left(-E_r^T \cdot A - W_{ijk} \right) \quad \forall i, j, r \quad (2)$$

and A is the average face image. Since all values needed to compute C 's are available after the training phase is done, C 's can be computed offline.

The design for the architecture to perform face recognition would utilize both parallel and pipelining techniques. Inherent parallelism in pixel-based computations of the summation of products between the eigenvectors and the sub-image is exploited fully. Once all the sums are available, the computation of the minimum distance is carried out independently of the parallel procedure just described; therefore, these two steps are implemented as two stages of a pipelined

architecture. The first step is to perform the computation of

$$P_r = E_r^T \cdot I_{test} + C_{ijk} \quad \forall r \quad (3)$$

and the second step is described as:

$$D_{ijk} = \sum_{r=1}^{M'} |P_r| \quad (4)$$

Figure 1 shows the pipelined architecture that implements the two main computational tasks described above.

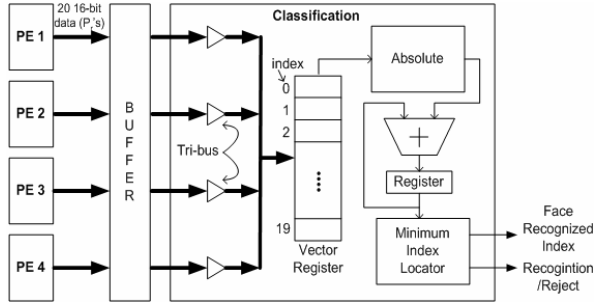


Figure 1. Block diagram for the face recognition system

Each processing element (PE) in Figure 1 is designed to perform the computation task of (3) and each PE has 20 parallel processing lanes corresponding to 20 eigenvectors used in the training phase. The computational procedure in the classification module involves simple operations of computing the absolute values for all P 's, accumulating them and determining the minimum distance. Since each PE can produce results for 20 P 's and the results of each PE are available every 257 cycles, more PE's are needed to operate in parallel to provide sufficient data so that the classification module would not be idle most of the time. Based on the required performance and the available resources in the targeted FPGA, 4 PE's are designed to operate in parallel as shown in Figure 1.

3. Experimental Results

The architecture design presented here is implemented with VHDL utilizing Altera's Quartus synthesis software. The designed architecture is fitted in an Altera's EP20K600CB652C7 FPGA in which 80% of the chip's logic elements are consumed. The architecture is capable of operating with a 91 MHz clock rate. The results for performance analysis of databases with different sizes are listed in Table 1 where DB120 indicates the database with 120 images. Likewise, DB1000 and DB10000 indicate database with 1000 and 10000 images respectively. The face recognition system presented in this paper is a flexible design, which can be expanded by connecting more

PE's in parallel to perform face recognition in real-time for very large database (with 10000 or more images).

Table 1. Recognition time of the architecture with respect to different face image databases

Database	Recognition Time (ms)
DB120	1.4
DB1000	11.3
DB10000	113.1

4. Conclusion

A face recognition system based on the modular PCA method was designed and implemented. The high throughput of the face recognition system was achieved by using a multilane architecture design approach which consists of 4 PE's and a classification module. In addition to PE's being operated in parallel, there are 20 parallel computational paths/lanes in each PE to perform calculation on 20 eigenvectors simultaneously. PE's and classification module are integrated in a pipelined fashion to provide parallelism in sequential operations. The architecture was able to perform face recognition in 11ms for a database with 1000 face images. The proposed architecture can handle real-time face recognition even for very large face databases by connecting more PE's in parallel.

5. References

- [1] Gottumukkal R., and Asari K.V., "An improved face recognition technique based on modular PCA approach," *Pattern Recognition Letters*, vol. 25, no. 4, 2004, pp. 429-436.
- [2] Chellappa, R., Wilson, C. L., and Sirohey, S. "Human and machine recognition of faces: a survey," *Proc. IEEE*, vol. 83, No. 5, May 1995, pp. 705-740.
- [3] Brunelli, R., and Poggio, T. "Face recognition: feature versus templates," *IEEE Trans. Pattern Anal. Machine Intell.* vol. 15, no. 10, 1993, pp. 1042-1052.
- [4] Turk, M., and Pentland, A. "Eigenfaces for Recognition," *Journal Cognitive Neuro-science*, vol. 3, no. 1, 1991, pp. 71-86.
- [5] Gottumukkal R., and Asari K.V., "System Level Design of Real Time Face Recognition Architecture Based on Composite PCA," *Proc. GLSVLSI 2003*, 2003, pp. 157-160.
- [6] Ngo H. T. and Asari K. V., "A pipelined architecture for real-time correction of barrel distortion in wide-angle camera images," *IEEE Trans. of Circuits and Systems for Video Technology* vol. 15, no. 3, , March 2005, pp. 436-444.