13th Euromicro Conference on Digital System Design: Architectures, Methods and Tools, September 2010.

Hardware-Based Speed Up of Face Recognition Towards Real-Time Performance

I. Sajid and Sotirios G. Ziavras
Electrical and Computer Engineering Department,
New Jersey Institute of Technology, Newark, NJ 07102, USA

M. M. Ahmed
Department of Electronic Engineering,
Mohammad Ali Jinnah University, Islamabad, Pakistan

Abstract— Real-time face recognition by computer systems is required in many commercial and security applications because it is the only way to protect privacy and security in the sea of peoples. On the other hand, face recognition generates huge amounts of data in real-time. Filtering out meaningful data from this raw data with high accuracy is a complex task. Most of the existing techniques primarily focus on the accuracy aspect using extensive matrix-oriented computations. Efficient realizations primarily reduce the computational space using eigenvalues. On the other hand, an eigenvalues oriented evaluation has minimum time complexity of O (n^3), where n is the rank of the covariance matrix; the computation cost for co-variance generation is extra. Our frequency distribution curve (FDC) technique avoids matrix decomposition and other high computational matrix operations. FDC is formulated with a bias towards efficient hardware realization and high accuracy by using simple vector operations. FDC requires pattern vector (PV) extraction from an image within O (n^2) time. Our enhanced FDC-based architecture proposed in this paper further shifts a computationally expensive component of FDC to the offline layer of the system, thus resulting in very fast online evaluation of the input data. Furthermore, efficient online testing is pursued as well using an adaptive controller (AC) for PV classification utilizing the Euclidian vector norm length. The pipelined AC architecture adapts to the availability of resources in the target silicon device. Our implementation on an XC5VSX50t FPGA demonstrates a high accuracy of 99% in face recognition for 400 images in the ORL database, generally requiring less than 200 nsec per image.

Keywords-component: Real-time, Face Recognition, Reconfigurable.

I. INTRODUCTION

Intrusive and non-intrusive are the main categories of biometric techniques. Researchers for defense, security and commercial applications increasingly demand face recognition that employs non-intrusive biometric approaches, thus protecting both security and privacy [1]. Such systems generate a lot of raw data in real-time. Numerous engineering applications that make a decision based on raw data employ pattern recognition approaches that first extract meaningful data [2] and then generate eigenvalues to represent this dataset. Furthermore, PCA (Principal Component Analysis) is such a common technique associated with face recognition algorithms [2] [3] [4] [5] [6] [7] [8] [9].

PCA computes the eigenvalues, usually in $O(n^3)$ time, where n is the rank of the co-variance matrix [20]. PCA provides less than 85% accuracy even when using 50% of the training images per subject [10] [11] [12] [13] [14]. Many variants of PCA improve the accuracy and/or speedup the computation [15] [16]. Linear Discriminant Analysis (LDA) normally provides better accuracy than PCA when the dimensionality of the transformed space is one less than the classes used in the training set. Furthermore LDA needs more than 80% of the stored images per subject for training purposes [13] [14]. In fact, LDA's accuracy deteriorates badly when the system is trained with only one or two images per subject in the training set. Furthermore, it has higher computational cost than PCA because it uses PCA along with the multivariable normal distribution of the covariance matrix.

Fisher Discriminant Analysis (FDA) works well with the reduced space created by PCA. FDA uses LDA to optimize the sample data point projections [13] [14]. FDA and LDA use a global Euclidian structure for the extraction of features and ignore local face details. Therefore, FDA may not achieve 99% accuracy that is frequently required for reliable real-time processing. Real-time pattern recognition needs algorithms that can extract and identify known features in minimum time [29]. Relevant embedded systems are expected to recognize faces in a fraction of a μsec. Such a real-time system proposed by Microsoft can handle 15 frames, with two frames of face detection per second, when implemented on a 700 MHz Intel or a 200 MIPS-strong ARM processor [17]. Any software-based application realization is normally slower compared to its hardware-based counterpart. For example, hardware-based face recognition using an artificial neural network and eigenfaces has been implemented on an analog ADSP-BF535 EZ-KIT device [18]. This system provides recognition with a maximum accuracy of 80%, consumes 36 msec each time, and uses more than one MB of storage for each face. A multi-processor architecture that includes a smart camera achieves recognition in 4.3 msec with 90% accuracy [19].

Many algorithms have been developed to achieve maximum accuracy within minimum time [10] [20] [11] [21] [22] [23]. However, none of them demonstrates ~99% accuracy within $O(n^2)$ time with a single training image per subject, for popular benchmark databases of faces. Our proposed technique extracts both global and local details using the simple FDC matching method that requires storage space for $m \times 256$ array of gray levels on the host machine and two vectors simultaneously in the BRAM

memory of the FPGA, where *m* is the number of subjects. Our herein proposed technique needs only one training image per subject, with frontal pose and good lighting, to provide reliable recognition within $O(n^2)$ time. Accuracy may reduce in case training images are acquired with bad lighting or angled pose, but decision will be provided within $O(n^2)$ time.

Hardware realization can be customized considering FPGA resource constraints in memory and other on-chip areas that can be configured for computations. With our FPGA-based architecture, the generation of PVs, a computationally expensive process, is accomplished offline. These PVs are then grouped at static time according to their similarities in order to intelligently minimize the online recognition time by focusing at runtime only on those PVs that have a high probability to match the input. An adaptive controller in the online layer speeds up the recognition rate very substantially. It is shown here that the architecture with the adaptive PV-Controller (APVC) improves the recognition rate of the basic, non-adaptive architecture by 80%. Furthermore, the APVC-based pipelined architecture can be configured to match available device resources and application time requirements.

II. EIGENVALUE METHODS

Let $m_1, m_2, \ldots, m_d$ be the square matrices of rank *n* representing the training images for a subject. Assume the transformation $m_i \rightarrow x_i$ of each training image i into a column vector representing the original n^2 pixels. Then define a matrix $A = \{x_1, x_2, \ldots, x_d\}$, *for i=1, 2, …, d, where* each column of *A* represents an image and belongs to one of *m* classes $\{X_1, X_2, \ldots, X_m\}$. Let us also consider a linear transformation mapping the original $d*n^2$-dimensional image space of *A* into a reduced *d*-dimensional feature space using PCA (as described below).

A. PCA Reduced Eigen Spaces

The basis vectors are computed in the reduced dimensional space for the eigenfaces. The eigenvectors of the co-variance square matrix generate an Eigen space. The eigenvectors are denoted by $\mu_j, for\ j = 1,2,3 \ldots, d$. These vectors are computed usually by tridiagonalization and subsequent decomposition of the co-variance matrix [15]. The co-variance matrix is computed as $C = A^T A$, and the basis vectors for the $n^2 xd$ dimensions are computed with an original space of $n^2 xn^2$, that is

$$t_j = \frac{Au_i}{\|Au_i\|} \ and\ then\ T_{n^2 d} = [t_1\ t_2 \ldots . t_d\ ] \ldots\ldots (1)$$

Then, the projection of each of the *L* leading eigenvectors is computed and stored in a matrix $W_{d \times h}$ as follows:

$$W(i, j) = A_i^T\ t_j\ ;\quad i = 1,\ 2,\ 3,\ .....,\ d \quad \& \ j = 1, 2, 3, ..., L \qquad \ldots\ldots (2)$$

Let N be a column vector of length n^2, representing an input image. Define three column vectors D_N, W_N, V_N, as follows:

$$D_N = N - J \qquad \ldots\ldots. (3)$$

where *J* is the row mean of the *A* matrix having dimensionality $1 \times n^2$. The following equation represents the projected space and Equation (5) represents an error vector whose entries are associated with an input test image.

$$W_N(j) = (D_N)^T t_j;\quad j = 1,\ 2,\ .....,\ L \qquad \ldots. (4)$$

$$V_N(i) = \left\| W_N - (i^{th} column\ of\ W^T) \right\|;$$

$$i = 1, 2, 3, ..., d. \qquad \ldots\ldots. (5)$$

Equation (5) is part of the PCA classifier, after a statistical scalar value is computed in relation to each input test image followed by threshold error limit analysis to achieve recognition. However, the underlying PCA reduction process requires O (n^3) minimum time.

B. Fisher Faces

Eigenfaces maximize the variance between classes while ignoring the within-class variance. Fisher faces use LDA to compute both variances separately to seek the direction of an efficient discrimination between classes [12] [24]. Fisher faces usually perform better than LDA and PCA when data for the classes are uni-modal and the training data set includes a large number of images per subject [13] [14].

Let K_b and K_w be the variance between classes and within a class, respectively:

$$K_b = \sum_{i=1}^{m} (u_i - u)\ (u_i - u)^T \qquad \ldots\ldots\ldots (6)$$

$$K_w = \sum_{i=1}^{m}\ \sum_{j=1}^{N_j} \left(X_j^i - u_i\right)\ \left(X_j^i - u_i\right)^T \qquad \ldots\ldots. (7)$$

where N_j is the number of images in class j, *m* is the total number of classes, and X_j^i is the j^{th} sample in the i^{th} class. An optimal projection can be obtained by maximizing the ratio of the determinant in Equation (8):

$$W_{LDA} = \arg \max_W \frac{|W^T K_b W|}{|W^T K_w W|} \qquad \ldots\ldots\ldots\ldots (8)$$

$W_{LDA} = [W_1, W_2, \ldots., W_L]$ contains the *L* largest eigenvectors. Projections between classes can be derived using Equation (7) and the corresponding eigenvalues [24]. However, LDA uses supervised learning and focuses on global structure because the denominator in Equation (8) is to be minimized or the numerator to be maximized.

III. PROPOSED FACE RECOGNITION ALGORITHM

Various pattern classifiers have been applied to face recognition, such as nearest neighbor, Bayesian, and support vector machine [3] [21] [25] [26]. However, they do not show a close to 100% accuracy while they consume $O(n^2)$ time with just one or two training images from three popular benchmark face databases. In this paper, we develop a new classifier based on cumulative frequency distribution and use the standard variance vector (SVV). SVV acts as a template kernel and is represented by a graph.

Figure-1 shows the plots of the cumulative frequency against the gray levels. Two same class images and three other class images are used for the proposed transformation. We can observe that plots belonging to the same class do not deviate substantially in Figure-1. The graphs are divided into three regions using three threshold values. It has been observed that the exclusiveness of the three regions provides better decision quality. Further, it cannot increase the search time by more than *O (n)* and the computing time for an input image is not more than *O* (n^2). The input image has to pass through the same computing steps which were used to extract the features in the training process. The proposed classifier provides linear time decision making for the input test image.

Let $T = \{x_1, x_2, \dots, x_m\}$ be a training set having m classes/columns. Define a transformation P on T which computes the gray level distribution as follows:

$$P\left(x_{ij}\right) = n_{ij}, \text{ where } i = 1,2, \dots, m \text{ and } j = 0,1,2, \dots,255 \quad \dots (9)$$

$$\text{We then normalize the distribution: } Pn_{ij} = \frac{n_{ij}}{rc} \quad \dots (10)$$

where rc defines the resolution of the images. We accumulate the distribution as calculated in Equation (9) and then do sorting which provides a gradually increasing order of the frequencies of gray level appearance. Each row in the normalized Pn matrix represents a reference for the respective class.

Let Pt be a transformed testing PV vector obtained by applying Equations (9) and (10) to the incoming test image, which is then subtracted from a column of the Pn matrix to populate a vector *M*. The content of the M vector is over written for each input test image, because the decision entry is simultaneously stored in the *U* vector.

$$M_{iN} = Pn_{ij} - Pt_{Nj,}$$
$$where\ N\ is\ the\ number\ of\ test\ images \quad \dots . (11)$$

To maximize the variance between classes, each vector in Equation (11) is divided into three regions because the distribution in Equation (11) shows Gaussian behavior [27]. The minimization of the following objective function, which has been developed for efficient face recognition, provides a better discrimination analysis:

$$U_{ij} = \begin{cases} 1 & std(M_{iN_{j1}}) < \varepsilon_1 and\ std(M_{iN_{j2}}) < \varepsilon_2\ and\ std(M_{iN_{j3}}) < \varepsilon_3 \\ 0 & otherwise \end{cases}$$

$$Where\ \varepsilon_{1,}\ \varepsilon_2,\ \varepsilon_3 \in \{0,1\} \quad \dots . (12)$$

It is observed that distinguishing among three regions in Equation (12) for the variance vector improves the discriminating power of the proposed technique. The variance vector has a critical role as observed in Figure-1. Furthermore, $the\ j1, j2, j3$ lengths have to be defined in such a way that the computed thresholds can be compared with $\varepsilon_{1,}\ \varepsilon_2,\ \varepsilon_3$, respectively It has been observed that *j1+j3 <= j2* and ε_3 provides better results under the constraint $j1 + j2 + j3 = 255$ [27]. TrIM1 TrIM2 and TrIM3 represent the FDCs of randomly selected images from the same class in Figure-1. While TestIm1, TestIm2 and TestIm3 represent FDCs, they are for randomly selected images from other classes in the ORL face database (used in our experiments) [28].

IV. Architecture

Our FDC-based three-layer architecture is shown in Figure-2. The feature extraction for the set of training images using the proposed FDC technique is a computationally expensive process that is handled offline during pre-processing. This process produces the PVs in time depending upon the number of subjects *d* in the given data set. The PVs are stored in the host machine's RAM memory. One PV is transferred at a time to the online layer through the PCI or other bus interface. The number of pre-stored images per subject determines the total test space.

The second layer of the architecture is used for online testing. The digital image, which is treated as a matrix of gray levels, requires substantial bandwidth to be transferred to the FPGA. Such a bandwidth may not be available due to their rather limited number of input/output pins and their operating frequencies. For this reason, our approach here converts the image matrix into the one-dimensional input pattern vector (IPV) on the host machine for transmission to the FPGA board. An FPGA buffer directs this data to the on-chip BRAM memory for efficient FDC computation. FDC applies the mathematical steps in Equations (10) and (11).

One PV from the pre-stored collection of PVs in the host RAM is transferred while the IPV is being computed, thus overlapping computations with data transfers. After the IPV is computed in the FPGA, the classifier determines statistically, as per Equations (11) to (12), a possible match (binary decision). This process repeats for all PVs pre-stored in the host RAM unless a true decision is made by the classifier. Our experiments show that, on the average twelve iterations are needed for a successful recognition. We further improve the efficiency of the architecture in Figure-2 by introducing an adaptive classification technique for the pre-stored (training set based) PVs.

The enhanced architecture relies on a process that eliminates the need to test all the pre-stored PVs against the online produced IPV. The proposed architecture modifications are minimal. The new architecture is shown in Figure-3. In this architecture, the Euclidian normalization length of a vector is proposed to calculate tags for the pre-stored PVs. The tags are scalar real values computed for the PVs. The objective is to minimize the number of false attempts by introducing the following algorithm in the host machine.

Algorithm for the controller to locate the most suitable PV for the incoming test image

Step (1): Using the Euclidian normalization length formula $x = \left[\sum_{i=1}^{256}\left(abs\left(x_i^2\right)\right)\right]^{1/2}$ calculate tags for the stored PVs.

Step (2): The computed tags for PVs are stored in a vector $V_m = x_j$, *where* $1 \leq j \leq$ *m and m is t e number of subjects in a database*

Step (3): Apply step (1) for the test image and get the *y* vector, and subtract *y* from each element of V_m to generate a Euclidian length difference vector *Diff_PV* of dimension *m*.

Step (4):Sort the *Diff_PV* in ascending order and store the difference values with their indices in another array *F_PV* having dimension *2 x m*.

Step (5): The index associated to the first value in *F_PV* is used to send the first PV having the maximum probability to match with the IPV. If recognition is not successful, then choose the next value from F_PV and repeat this process until a successful recognition signal is received from the next hardware layer.

Step (6): Repeat steps 3 to 5 for each test image.

In Figure-3, the host layer has a tag vector of length *m* generated during step (2) of the above algorithm. In the online layer, two signals are introduced to control the next transfer of the *PV* from the host layer to the online layer. The signal TF_Fag has a "true" value for a matching and "false" otherwise (i.e., demanding the next PV). A "true" value for TF_Fag is also transmitted to the I/O butter to get a new test image. Furthermore, for a "true "signal, a new ID_IPV is sent to the host layer.

Our simulation results in histogram form for the new algorithm using the ORL database are shown in Figure-4 [28]. The column length indicates the frequency of successful attempts whereas the horizontal axis shows the number of required tests against PVs. The proposed controller algorithm provides significant improvement for face recognition compared to the original architecture in Figure-2. The first column in Figure-4 shows that ~190 test images will get a successful PV from the PV_controller within the fifth iteration. The average number of run-time tested PVs is reduced to 7.7 from 20.

A. Pipelining

In Figure-5 a pipelined architecture is formulated to take advantage of regularity in *PV_ID*, *IPV_ID* and *Vm* array data. In this architecture, PV_controller is replaced with a new layer for PV classification. This layer has three major components, namely Trained_PV, P_PV_controller and PV_Cache. Trained_PV isrepresents the collection of PVs, similar to the architecture without the pipeline. P_PV_Controller has two extra components compared to the architecture in Figure-3. These components are Circular FIFO and a *k*-dimensional array containing the sorted tags of PVs in the corresponding pipelined modules. Circular FIFO is introduced in the PV_Controller to handle the sequence of identification tags for incoming test images. It is assumed that the *k* modules can get their most suitable trained PVs from the controller layer; these PVs are stored in advance in the PV_Cache area.

The diamond block provides input to the OR gates to preempt early any further processing after a successful match/recognition. This block generates a positive decision if the test image belongs to a person in the training database; otherwise, its feedback signal to the host machine requests the next appropriate PV. This block plays an important role in the pipelined architecture using the PV_Cache area, as shown in Figure-5.

A "true" or "1" answer for the Boolean expression in Equation (12) means that the input image matches that particular PV; otherwise, the next appropriate PV has to be sent from PV_cache to the online layer. PV_Cache holds the PVs according to the number of test images being under process in the online layer. PV_Cache will be flushed when a "true" signal is received from one of the OR gates.

The horizontal modules in the online layer in Figure-5 depend upon the target FPGA resources and the vertical elements determine the number of FPGAs on the target board. Therefore, the proposed pipelined architecture is a modular, robust reconfigurable system as shown in Figure-6. The obtained architecture speed as a function of the pipeline stages was simulated and the result is shown in Figure-6, assuming a 10% overhead for new stages.

The output of the OR gates that generate a feedback to the controller further improves the decision time for an appropriate PV identification that leads to a transmission to the online layer. A high speed face recognition process can then be obtained.

V. RESULTS

Table-1 shows the resource utilization data and the execution time when implementing our architectures on a Xilinx Virtex5 FPGA device. The Xilinx 11.1 suite was used that includes the AccelDSP tool with MATLAB files for floating-point verification. According to the synthesis report, 15 32-bit multipliers, 53 adders and 47 subtractors are used in the implementation. Pipelining in the multipliers, adders and subtractors improves the time further but the resource consumption reaches 36% of the FPGA real estate. On the other hand, an approximate 80 nsec execution time makes our system a very viable choice for real-time face recognition. Our chosen development tool provides two frequencies for the designed system, one being the requested frequency and the other the maximum based on the circuit's critical path after RTL implementation. The execution time for both frequencies is shown in Table-1 using the ORL database. The worst frequency is the selected value before the synthesis process mentioned in Table-1. The pipelined architecture shows an improvement of more than 87 % compared to the architecture in Figure-3 for the XC5VSX50t Virtex5 device since up to three modules in Figure-5 can work simultaneously. The testing of any image using the proposed system provides a decision within 0.6 µsec with accuracy of 98.3%. This was validated with the ORL database of 400 images with 40 subjects and 5600 pixels in each gray level image. While Microsoft real-time

system provides decision in 0.5 sec and analog system gives result in 36 msec with 80% accuracy.

VI. CONCLUSION

Real-time face recognition systems have very high computation demands while also requiring high accuracy. These requirements are more vital when dealing with security applications. Most of the existing face recognition algorithms have been developed for desktop-based offline systems. Our proposed frequency distribution curve matching technique primarily pre-calculates pattern vectors (PVs) that can be subsequently used by the online classifier. Novel FDC-based architectures were presented that achieve substantial speedups while also providing highly accurate recognition. The adaptive PV_controller-based FDC architecture yields high speedups with due consideration to resource constraints stemming from the chosen FPGA device. Furthermore, the pipelined architecture exploits the parallelism capabilities of configurable devices, thus providing even more viable solutions for real-time face recognition tasks.

Table-1. Resource consumption and performance for FDC-based face recognition

ORL face database with 40 subjects	Utilization	
FPGA Device: Xilinx XC5VSX50T	Resources	
Slices	36%	
BRAM	1%	
Multipliers	15	
Adders	53	
Subtractors	47	
DSP units	19%	
Time of Execution	Best	Worst
Frequency in MHz	188	100
Data feeding Time (ns)	118.72	224
Binary decision data out time (ns)	0.0053	0.01
Test time for a pattern vector (ns)	126.39	233.97
Test time for the ORL database (ns)	2527.71	4679.40
Time for online testing using the adaptive controller (ns)	973.17	1801.57
Time for the pipelined architecture (ns)	421.70	780.67
Improvement with pipelined architecture % (100-(PA/AC)*100)	83.316	83.316

PA= Pipelined architecture, AC= Adaptive controller architecture

REFERENCES

[1] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face Recognition: A Literature Survey," ACM Computing Surveys, vol. 35, no. 4, Dec., 2003.

[2] S. Yamada, A. Nafalski, S.C. Mukhopadhyay, "Dynamic image cognition along with eigen patterns", Application of electromagnetic phenomena in electrical and mechanical systems, 2003.

[3] M.A. Martinez and A.C. Kak, "PCA versus LDA," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 23, no. 2, pp. 228-233, Feb. 2001, doi: 10.1109/34.90897. (IEEE Transactions)

[4] H. Xiaofei, Y. Shuicheng, H. Yuxiao, N. Partha and Z. Hong-Jiang, "Face Recognition Using Laplacian Faces," *IEEE Trans. on Pattern Analysis and Machine Intelligenc,* vol. 27, no. 3, pp. 328-340, Mar. 2005, doi: 10.1109/TPAMI.2005.55. (IEEE Transactions)

[5] W. Zhao, R. Chellappa, P.J. Phillips and A. Rosenfeld, "Face Recognition: A literature survey," *ACM Computing Surveys (CSUR),* vol. 35, no. 4, pp. 400-459, 2003.

[6] R. Chellappa, C.L. Wilson, S. Sirohey, "Human and Machine Recognition of Faces: A Survey," *Proc. IEEE* vol.83, no. 5, pp. 705-740, May1995, doi: 10.1109/5.381842. (IEEE Conference proceedings)

[7] J. Daugman, "Face and Gesture Recognition: Overview," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 7, pp. 675-676, Jul. 1997, doi: 10.1109/34.598225. (IEEE Transactions)

[8] S. Pankanti, R.M. Bolle and A. Jain, "Biometrics: The Future of Identification," *Computer,* vol. 33, no. 2, pp. 46-49, Feb. 2000.

[9] A. Pentland, T. Choudhury, "Face Recognition For Smart Environments," *Computer,* vol. 33, no. 2, pp. 50-55, Feb. 2000.

[10] D. Cai, X. He, J. Han and H. J. Zhang, " Orthogonal Laplacianfaces for Face Recognition", *IEEE Trans. on Image Processing*. 15(11) (2006).

[11] X. He, S. Yan, Y. Hu, P. Niyogi and H. Zhang, "Face recognition using laplacian faces", *IEEE Trans. on Pattern Analysis and Machine Intelligence*. 27 (3) (2005) 328-340.

[12] H. Hu, "Orthogonal neighborhood preserving discriminant analysis for face recognition", *Pattern Recognition*. (2007) 2046-2054.

[13] A. Eleyan and H. Demirel, "PCA and LDA based face recognition using feed forward neural network classifier", LNCS 4105, 2006, pp. 199-206.

[14] M.A. Martinez and A.C. Kak, "PCA versus LDA," *IEEE Trans. Pattern Analysis and Machine Intelligence.* 23(2) (2001) 228-233.

[15] A. Sharma, K.K. Paliwal, "Fast principal component analysis using fixed-point algorithm", *Pattern Recognition Lett.* 28 (2007) 1151-1155.

[16] I. Sajid, M. M. Ahmed and M. Sagheer, "An Improved Face Recognition System Using Stable Orthogonalization in Eigenspace", *Submitted in IEEE Trans. Image Processing*.

[17] P. Viola and M. J. JONES, "Robust Real-Time Face Detection", International Journal of Computer Vision 57(2), 137–154, 2004.

[18] M. Wei and A. Bigdeli, "Implementation of a Real-Time Automated Face Recognition System for Portable Devices", *lntwimtioiial Symposium on Communications and Infom~ation Technologies 2004 (ISCIT 2004) Sappom, Japan*, Octoher 26- *29,* 2004.

[19] H. Fatemi, R. Kleihorst, H, Corporaal and P. Jonker," Real-time face recognition on a smart camera", Proceedings of Acivs 2003 (Advanced Concepts for Intelligent Vision Systems), Ghent, Belgium, September 2-5, 2003.

[20] M. Turk and A. Pentland, "Eigenfaces for recognition". *J. of Cognitive Neuroscience.* 3 (1) (1991) 71-86.

[21] B. Moghaddam, and A. Pentland, "Probabilistic visual learning for object representation," *IEEE Trans. Pattern Anal. Mach. Intell*. 19 (7) (1997) 696–710.

[22] L. Z. Liao, S.W. Luo, and M. Tian, "Whitenedface: recognition with PCA and ICA", IEEE *Trans. of Signal Processing Letters.* 14 (2007).

[23] K. I. Kim, K. Jug and H. J. Kim, "Face recognition using kernel principal component analysis", *IEEE Signal Processing letter*. 9 (2) (2002)

[24] P. N. Belhumeur, J. P. Hespanha and D. J. Kriegman, "Eigenfaces vs. fisherfaces: recognition using class specific linear projection," *IEEE Trans. on Pattern Analysis and Machine Intelligence.* 19 (7) (1997).

[25] R. O. Duda, P. E. Hart and D. G. Stork, "Pattern Classification", John Wiley & Sons, 2001.

[26] P. J. Phillips, "Support vector machines applied to face recognition," in Advances in Neural Information Processing Systems. Cambridge, MA: MIT Press, 1998, pp. 803–809.

[27] I. Sajid, M.M. Ahmed, S.G. Ziavras, "Efficient Face Recognition Using Frequency Distribution Graph Matching", International Journal

of Pattern Recognition and Artificial Intelligence (IJPRAI), .(Submitted for publication)

[28] http://people.cs.uchicago.edu/~dinoj/vis/orl/, 2009.

[29] K. Ricanek Jr., M. Savvides, D.L. Woodard and G. Dozier, "Unconstrained Biometric Identification: Emerging Technologies," IEEE Computer (Febr. 2010) 56-62.

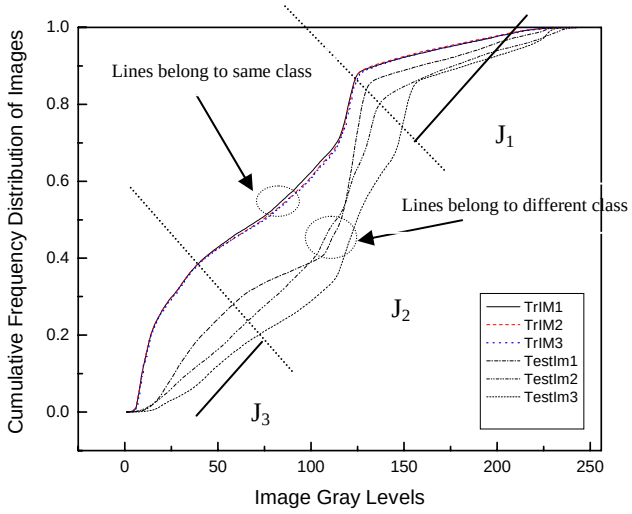

Figure1. Cumulative frequency distribution of gray levels for three images from the same class and three from other classes against gray levels for the ORL database.

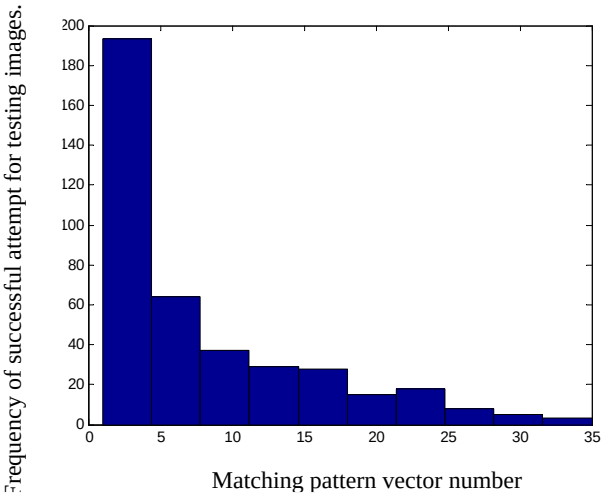

Figure -4. Simulation results for the proposed classification algorithm with pattern vectors using the ORL database.

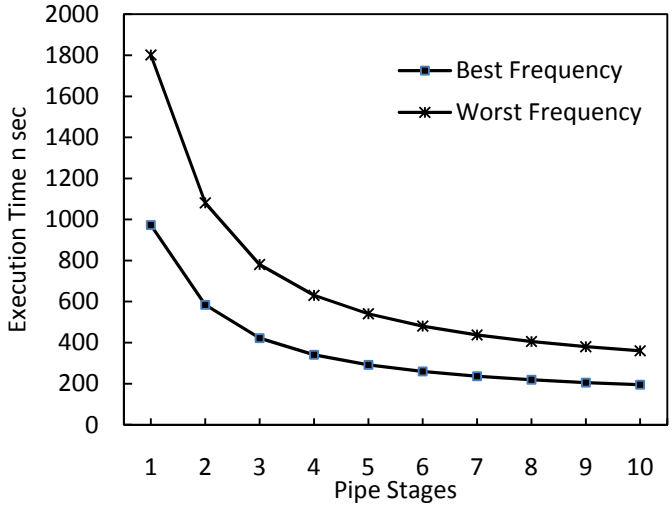

Figure 6. Pipelining performance versus the number of stages.

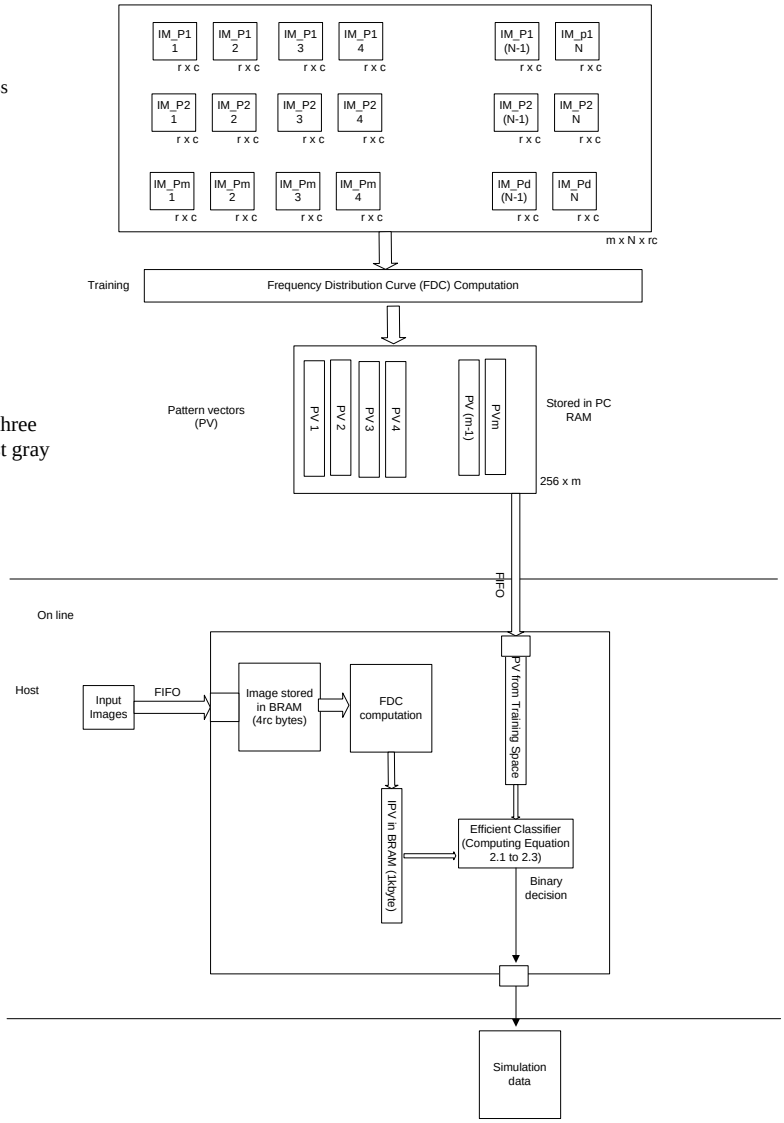

Figure 2. FDC-based face recognition architecture.

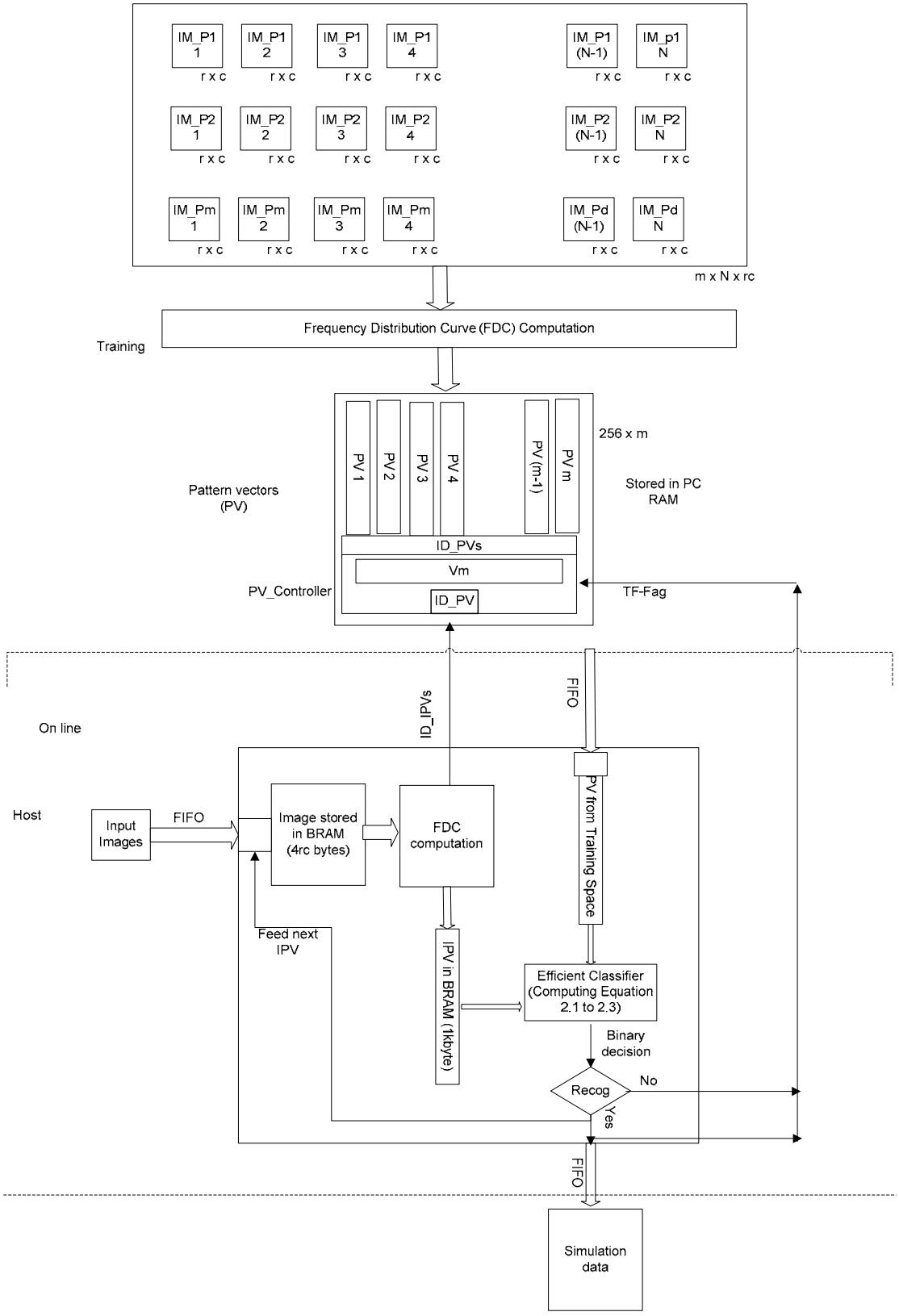

Figure3. FDC-based face recognition architecture using adaptive control for pattern classification

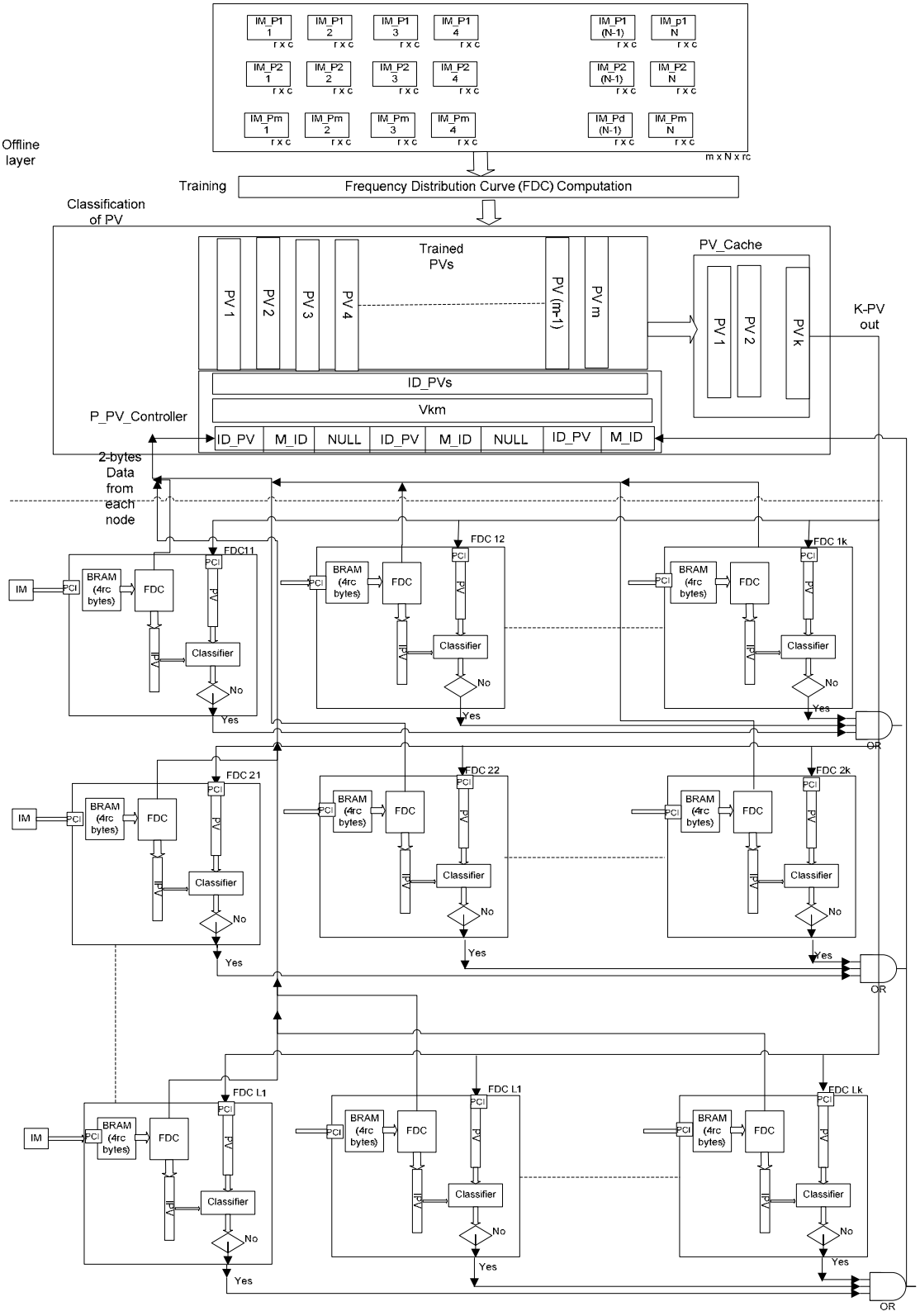

Figure 5. Pipelined FDC-based face recognition architecture.