TIME EFFICIENT FACE RECOGNITION FOR REAL-TIME APPLICATIONS

by

Imtiaz Ahmad Sajid

A dissertation submitted to M.A.J.U. in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY IN ELECTRONIC ENGINEERING

Department of Electronic Engineering
Faculty of Engineering and Applied Sciences
Mohammad Ali Jinnah University
December, 2010

DEDICATE TO

HOLY PROPHET MUHAMMAD (P.B.U.H)

THE GREATEST SOCIAL REFORMER

&

MY WORTHY PARENTS, SPOUSE AND KIDS

(FARZANA, HIRA, ABDULLAH)

ACKNOWLEDGEMENT

This dissertation describes research accomplished at the Microelectronic and FPGA design Laboratory, between February 2006 and August 2010. The project was carried out under the supervision of Professor Muhammad Mansoor Ahmed to whom I am delighted to express my sincerest thanks for his noble guidance, tremendous co-operation and continuous support of this interesting project. This project would not has been possible without his tutelage.

I would like to express my deepest gratitude to Dr. Muhammad Sagheer, for his enormous guidance and suggestion whilst pursuing this project. I would also like to thank Dr. Imtiaz Taj for many useful discussions and advises.

I have great regard to all colleagues in Computer Architecture and Parallel Processing Laboratory (CAPPL), Department of Electrical and Computer Engineering, New Jersey Institute of Science and Technology (NJIT), USA, they helped me great during my stay in CAPPL.

I would like to thank Mr. Waseem Ahmed, Dr. Anwar F. Chisti, Mr. Hussan M. Anwar and Mr. Arif Khatak for their encouragement and proof reading, which increased readability and reduced ambiguity in the thesis. It goes, without saying, that the advice, patience, and daily support of my colleagues of Microelectronic Laboratory.

I wish to express my sincere thanks to Dr. Saleem Asghar and his PhD students, Head of the Mathematics Department, Comsats Institute of Information Technology, Islamabad, they gave the literature which improves my work quality.

Finally, I must also thank the M.A.J.U administration for providing me an excellent environment for conducting research and the Higher Education Commission (HEC) for their financial support which geared the research work, and my parent Department which granted me the time to fulfill this work.

December, 2010

I.A. Sajid

List of Publication and Submission

1. I. Sajid, M.M. Ahmed, I. Taj, M. Humayun, and F. Hameed, "Design of High Performance FPGA Based Face Recognition System", PIERS Proceedings, Cambridge, USA, pp 504-509, July 2-6, 2008.
2. I. Sajid, M.M. Ahmed, and I. Taj , "Architecture of precise and time efficient implementation of Householder algorithm for face recognition on FPGA", dsd 08 Proceedings of work in progress, 11th euromicro conference on digital system design Architectures, Methods and Tools, Parma Italy, September 2-5, 2008.
3. I. Sajid, M.M. Ahmed, and I. Taj, "Design and Implementation of Face Recognition System Using Fast PCA", International Symposium on Computer Science and its Applications, pp 126-130, October 13-15, 2008.
4. I. Sajid, M.M. Ahmed, and I. Taj ," Time Efficient Face Recognition Using Stable Gram-Schmidt Orthonormalization", International Journal of Signal Processing, Image Processing and Pattern Recognition(ijsip), vol. 1, No. 02, March 2009.
5. I. Sajid and M.M. Ahmed,"Pipelined Implementation of fixed point Square Root in FPGA Using Modified Non-Restoring Algorithm", 2010 The 2nd International Conference on Computer and Automation Engineering, Feb, 2010, Singapore. (Accepted for publication).
6. I. Sajid, M.M. Ahmed and M. Sagheer" FPGA based optimized architecture for face recognition using fixed point Householder algorithm", 4th NISS: 2010 International Conference on New Trends in Information Science and Service Science, Korea, Gyeongju, 11-13 May, 2010.
7. I. Sajid, Sotirios G. Ziavras and M.M. Ahmed,"FPGA-Based Normalization For Modified Gram-Schmidt Orthogonalization", Internation Conference on Computer Vision Theory and Application (VISAPP), 17-21 May,2010, Angers, France.
8. I. Sajid, Sotirios G. Ziavras and M.M. Ahmed, "Hardware-Based Speed Up of Face Recognition Towards Real-Time Performance", 13th EuroMicro Conference on Digital System Design: Architecture, Method and Tools, lille, France, 1-3 September, 2010.
9. I. Sajid, M.M. Ahmed, Sotirios G. Ziavras, "Efficient Face Recognition Using Frequency Distribution Graph Matching", International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI), (Submitted for publication).

10. I. Sajid, M.M. Ahmed, Sotirios G. Ziavras," Novel Pipelined Architecture for Efficient Evaluation of the Square Root Using a Modified Non-Restoring Algorithm", The Journal of VLSI Signal Processing.

11. I. Sajid, M. M. Ahmed, M. Sagheer," FPGA-based Householder Algorithm Implementation for evaluation of Eigenvalues", International Journal of Innovative Computing Information and Control, (Second Revision).

12. I. Sajid, M.M. Ahmed, and M. Sagheer," An Improved Face Recognition System Using Stable Orthogonalization in Eigen Space", International Journal of Innovative Computing Information and Control, (Second Revision).

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACRONYM TABLE

AAM	Active Appearance Model
AC	Adaptive Controller
AccelDSP	Accelerating Digital Signal Processing
ALU	Arithmetic Logic Unit
API	Application Programming Interface
APVC	Adaptive PV-Controller
ARM	Advanced RISC Machines
ASIC	Application Specific Integrated Circuit
ASM	Active Shape Model
BRAM	Block RAM
BSS	Blind Source Separation
CIOB	Configurable I/O Block
CJA	CORDIC-Based Jacobi Algorithm
CLB	Configurable Logic Blocks
CMOS	Complementary Metal-Oxide Semiconductor
CORDIC	Coordinate Rotation Digital Computer
CRLB	Cramer-Roa lower bound
DB	Database
DC	Data Consumer
DLA	Dynamic Link Architecture
DP	Data Producer
DSP	Digital Signal Processing
EBGM	Elastic Bunch Graph Matching
ER	Error Rate
ERR	Equal Error Rate
FAM	Flexible Appearance Model
FAR	False Acceptance Ratio
FDA	Fisher Discriminant Anaysis
FDA	Fisher Discriminant Analysis
FDC	Frequency Distribution Curve
FIFO	First-in-first-out
FPGA	Field Programmable Gate Array
FPU	Floating Point Unit
FRR	False Rejection Ratio
GSO	Gram-Schmidt Orgothonalization
GW-chi	Gaussian-weighted chi-squared
HDL	Hardware Descriptive Language
HH	HouseHolder
HMM	Hidden Markov Model
ICA	Independent Component Analysis
ID_IPV	Identification of Input Pattern Vector
ILP	Instruction-Level Parallelism

ISOMAP	Isometric Mapping
IP	Intellectual Property
IPV	Input Pattern Vector
K-HLGPP	kernel learning of histogram of local Gabor phase pattern
KL	Karhunen-Loeve
LBP	Local Binary Pattern
LDA	Linear Discriminant Analysis
LEV	Leading Eigen Vector
LLE	Locally Linear Embedding
LPP	Locality Preserving Projection
LSA	Line Search Algorithm
LSB	Least Significant Bit
MAC	Multiply-Accumulate
MER	Minimum Error Rate
MFLOPS	Million Floating Operations Per Second
MGSO	Modified Gram-Schmidt Orthonormalization
MIPS	Million Instructions Per Second
MSB	Most Significant bit
MSE	Mean Square Error
NP	Non Pipeline
OLPP	Orthogonal Locality Preserving Projection
OPB	On-chip Peripheral Bus
ORL	Olivetti Research Laboratory
PCA	Principal Component Analysis
PCI	Peripheral Component Interconnect
PIE	Pose, Illumination, and Expression
PL_IPL	Pipelined with Instruction level parallelism
PLB	Processor Local Bus
PLB	Processor Local Bus
PV	Pattern Vector
RISC	Reduced Instruction Set Computer
RMS	Root Mean Square
ROC	Receiver Operating Characteristic
SPARC	Scalable Processor Architecture
SUN	Stanford University Network
SVM	Support Vector Machine
SVV	Standard Variance Vector
SYSACE	System Advanced Computing Environment
TF_Fag	True False flage
TS	Training Set
WAR	Write After Read
WPL	Without Pipelined
XC2VP2	Xilinx Virtex2Pro
XC3S100	Xilinx Spartan3
XE	Xilinx Edition

ABSTRACT

Biometrics play an important role in enhancing efficiency and reliability of authentication systems. Face recognition is a non-intrusive type of biometric technique that can be used in security applications where individual controlled image acquisition is not possible such as crowd, airport, stadium etc. Significant efforts have been made by the researchers to improve accuracy and efficiency of face recognition system to meet the stringent needs of defense, security and high-tech commercial applications. For real-time face recognition system in a dynamic environment where the face database may change continuously and learning is a reemerging process, continuous calculation of Eigen values and vectors is a primary requirement.

For a face recognition system, fast PCA is modified by incorporating Modified Gram-Schmidt Orthogonalization (MGSO) for efficient Eigen values generation. An adaptive classifier is developed for modified fast PCA, which provides an optimum selection of leading Eigen vectors. Further, a technique is proposed using two dimensional data structure array to achieve the best combination of system variables for its improved accuracy.

In a pattern recognition solution, it is important to acquire meaningful data by evaluating Eigen values which is one of the most computational intensive parts of the system. For hardware evaluation of Eigen values, coordinate rotation digital computer (CORDIC) based on Jacobi algorithm (CJA) is one of the best reported FPGA implementations. Contrary to CJA, Householder (HH) is considered an efficient method for Eigen solution. A co-design pipelined architecture is developed to implement HH by using FPGA. The accuracy of HH is further improved by evaluating square root using non-restoring algorithm. The proposed architecture demonstrated an improvement up to 30% in time and 10^{-7} decimal places in accuracy compared to CJA.

MGSO applies normalization of vectors in its iterative orthogonal process. Normalization involves square root and other arithmetic operations. Hardware realization of the floating-point square root operation might be prohibitively expensive because of its complexity. Three architectures have been developed that employ fixed-point hardware for efficient implementation of normalization of vectors on an FPGA. The application dependent suitability of these architectures is evaluated by using four popular databases.

High accuracy, with minimum decision time is a challenging task for a real-time face recognition system. PCA is a classical approach amongst the dimension-reduced feature extraction methods towards efficient recognition. However, PCA offers poor accuracy because it treats faces as global entities. The Local Preserving Projection (LPP) and Orthogonal Local Preserving Projection (OLPP) methods treat faces as combination of features; however, they are computationally intensive.

A technique based on Frequency Distribution Curve (FDC) is developed which also preserves global as well as local features but it avoids matrix decomposition and other

high order computational matrix operations. A software implementation of the technique showed an improvement up to 14%, 1.9% and 1.7% for the Yale, ORL and PIE databases respectively with relatively reduced training sets compared to PCA, LPP and OLPP.

FDC is formulated with a bias towards its hardware realization. An FPGA-based architecture has been developed by designing an adaptive pattern vector controller. This controller has the ability to detect high probability pattern vector for matching. The architecture performance is further improved by adopting parallelism which has demonstrated an improvement up to 80% compared to sequential operations. The system thus developed generally requires less than 200 ns to make a decision.

CHAPTER 1

INTRODUCTION

1.1 Introduction

Biometric features, like those of fingerprints, iris and face, can be used for authentication purposes in defense, security and commercial applications. These are natural features associated with humans and are unique for every person. Biometric technologies can be broadly classified into intrusive, where an interruption is required, and non-intrusive. Intrusive biometric usually provides higher accuracy than the non-intrusive (Sonkamble 2005). On the other hand, non-intrusive biometrics are more acceptable in authentication process due to inherent ease of image acquisition associated with them.

Face and voice recognition are non-intrusive biometrics. Face recognition is more flexible than voice recognition. But face acquisition process is complex because it needs better image quality with minimum variations due to change illumination conditions, head poses and facial expressions. Angle, scale and position variations to the nose, eyes, forehead, chin, cheek and mouth are undesired in a face image because the machine may perceive the changed face as a completely new face.

Since face recognition is non-intrusive, an acquisition system usually acquire images from a reasonable distance. Distant images usually have many other unwanted objects in the frame resulting in the requirement of face detection before the recognition process. For the past two decades a lot of research has been carried out to improve the

accuracy of automated face recognition (Chellappa 1995, Daugman 1997, Pankanti 2000, Pentland 2000, Martinez 2001, Zhao 2003, Xiaofei 2005).

Machine based systems treat images as 2D matrices or 1D vectors. The size of an image matrix depends upon the image resolution. Accuracy of an image recognition system thus requires the development of complex algorithms involving computationally heavy matrix/vector operations, such as matrix inversion or matrix multiplication. Therefore, face recognition algorithms are computationally intensive in nature.

Many algorithms have been developed to achieve low time-complexity using the concept of reduced face space (Chellappa 1995, Daugman 1997, Pankanti 2000, Pentland 2000, Martinez 2001, Zhao 2003, Shaknarovick 2004, Xiaofei 2005). Usually, original face space is reduced by a dimensionally reduction technique such as principal component analysis (PCA), that preserves most of the information of the original data and results in classification with reasonable accuracy. Real-time applications primarily require a time efficient algorithm with high accuracy, using small quantities of samples per subject. Hardware realization of such a system needs a co-design architecture with an optimal trade-off between time and power.

Field programmable gate array (FPGA) implementation is a tradeoff between a general purpose and an application specific integrated circuit (ASIC) solution. FPGA offers flexibility near to a general purpose solution whereas its performance is closer to an ASIC (Zafar 2005). It allows dynamic architecture and run time reconfigurability. It is cost effective for customized applications and is commonly used for prototyping which can then be converted into an ASIC (Deschamps 2006).

However, end products containing FPGAs are now common. FPGAs are often employed to create time and power efficient designs in order to facilitate real-time portable applications.

1.2 Biometric Technologies

Biometric is a technique used for identification of a person. It is usually based on one or more physical and behavioral characteristics. The physical characteristics could be hand geometry, iris pattern, fingerprints, voice and face structure. Variation of fingerprints with age relates to behavioral characteristic of a person and pitch of voice is various for different people relates to physical characteristics of a person (Sonkamble 2005, Delac 2004).

Broadly biometrics can be divided into intrusive and non-intrusive categories as shown in Figure-1.1. Intrusive type of biometrics needs interruption at the time of authentication like fingerprints verification or iris recognition. Interruption of each person for an authentication process is not practical in some situations like in a sports stadium. At the same time authentication is required for the security purposes. Voice and face recognition can be applied in a crowd without interruption. Voice recognition provides good results when voice signal is of good quality, but voice signal is not clear in a crowd. On the other hand, face images can capture in a crowd without interrupting them and then separate each face for recognition purpose. Therefore, face recognition can provide a reliable authentication process in a crowd without interruption.

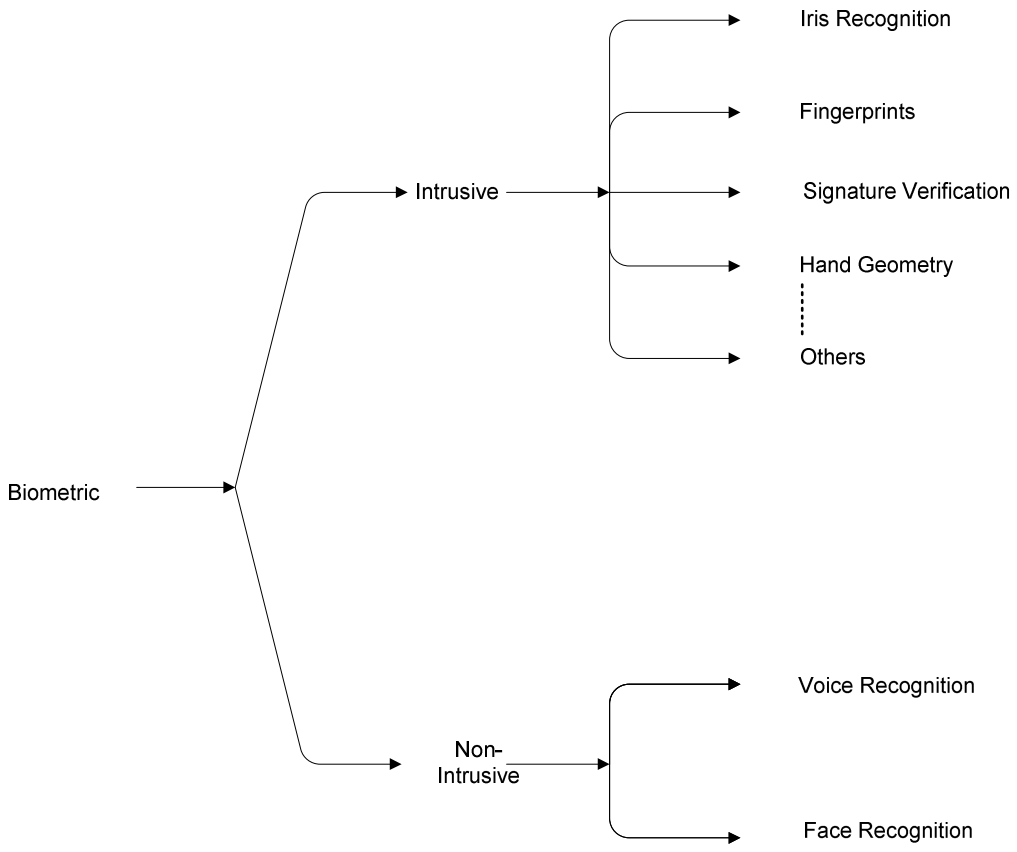


Figure- 1.1 Classification of biometric technologies based on type of biometric feature used.

1.3 Face Recognition System

A face recognition system should have the capacity of distinguishing a face among hundreds or thousands of other objects. Holistic face recognition starts after passing through three milestones.

One is face detection, second is to construct a projection matrix and the third is feature extraction (Kim 2002) as shown in Figure-1.2. In Figure-1.2, system is divided into off-line and on-line layers. Off-line layer shows block level explanation of training process. Training process starts after image acquisition and then face detection is carried out.

1.3.1 Face Detection

Face detection is mentioned as second block from top in Figure-1.2. Face detection is primarily concerned with the existence of a face in the given frame. In face detection, the image region is classified into face region and non-face region. An ideal system should have high true positive (detection at face region) and low false positive (detection at non-face region) detection rate. Face detection algorithms are categorized into knowledge-based methods, feature invariant approaches, template matching method and appearance based methods (Juell 1996, Leung 1996). Face detection depends upon facial features, image orientation and image quality (Yang 2002). One of the most famous works in face detection is that of Viola and Jones (Viola 2004) based on face/non-face learning of a number of weak classifier and then using AdaBoost for final classification. AdaBoost and its variants are the most commonly used face detection algorithms in commercial and practical systems. AdaBoost uses a small number of critical visual features of an image which provides a

fast face detection. Face detection is one of the mandatory pre-processing steps towards features extraction (Ming-Hsuan 2002).

1.3.2 Learning

In holistic algorithms, learning is a process of extracting the projection matrix from the training face images. A database of faces is first collected where each face is first pass through the first two steps of pre-process and detection. A subset of the database is chosen as training images such that it includes most of the variations of poses, illumination and age effect. The co-variance matrix of training images is calculated and the Eigen values and vectors are calculated as described by Turk et al (Turk 1991). Eigen vectors are the basis vectors or principal components of the projected face space. The projection matrix is constituted using the Eigen vectors corresponding to the highest Eigen values of the co-variance matrix as its rows. Eigen vectors are treated as extracted features of the given face space and Eigen vectors are used to construct a projection matrix. This projection matrix is used later for extraction of face features.

For a better recognition result, system usually needs large number of features and higher dimension projected matrix. A large quantity of features requires a big storage space and probably more time for recognition.

1.3.3 Extraction of Features

Features are extracted by multiplying projection matrix to each test image and to face in the original face database.

1.3.4 Projected Database

It is extracted from original face database after feature extraction of each image in the database. For recognition and comparison with test image this projected database is used.

1.3.5 Recognition

Extracted features are represented as data points usually in a face space. In general, recognition is based on a binary decision provided by a classifier. In classification process, input face image is projected into the extracted feature space. Projection matrix when multiplied with input face image gives a weight vector.

Weight vector and projected database are used to calculate distances of the input image into the projected space. These distances are treated as an error. This error is compared with a defined threshold value and then to decide that test input image is recognized or not.

Usually, classifier is a computationally intensive component of a face recognition system. On the other hand, it is responsible to provide decision within a limited time specifically for a real-time system.

1.4 Real-Time Face Recognition System

Figure- 1.3 represents a real-time face recognition. This figure is similar to Figure- 1.2 except the co-design and cache blocks. A real-time system has to respond within a deadline. This deadline may be a few seconds in some scenario and a fraction of nano second in other situations. In fact, deadline depends upon application nature. But it is

well known that hardware implementation reduces the response time of a system. There are two possibilities in hardware implementation of a system. The first one is that of a complete system implemented on hardware. This implementation can lead to a relatively time efficient system, however, it might be an expensive solution for a portable system. Second possibility of implementation is to divide the algorithmic code into two sections; one for the computational intensive parts and second for the rest of the algorithm. Computational intensive parts are implemented on hardware layer and rest of the algorithm on software layer. Such partitioning of the algorithm into software and hardware sections lead to a co-design methodology whose general concept is shown in Figure-1.3.

For design of dedicated hardware accelerator in a real-time system, the development cycle requires high density high performance reconfigurable media (Memik 2003) (Zafar 2005). FPGA is one of the best reconfigurable media for the development, validation and verification of an algorithm hardware accelerator (Kim 200, Saha 2007). A hardware accelerator based FPGA generally implements computationally intensive part of an algorithm.

Different vendors provide their own FPGA architectures. The basic architecture of FPGA consists of CLB, CIOB, interconnects to route signal between CLB and CIOB, master clock, BRAM and sometimes DSP units for multiplication and MAC operations (Brown 1996, Zafar 2005).

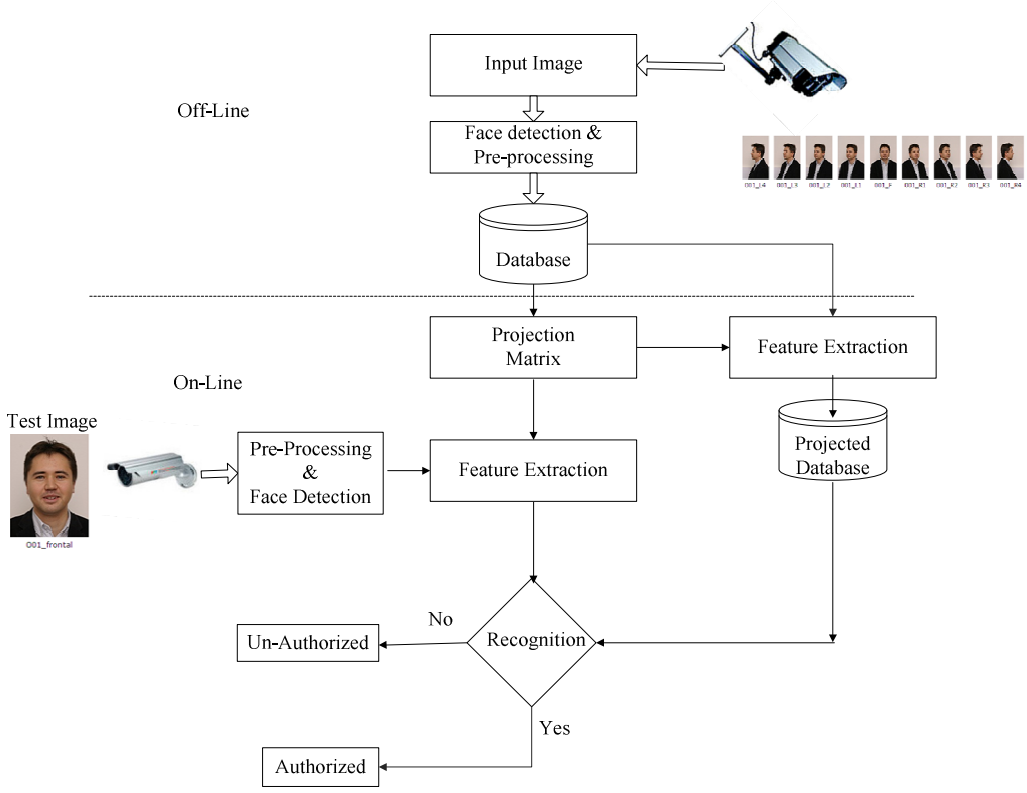


Figure- 1.2 A typical PC-based holistic face recognition invoking in dynamic environment.

1.5 Motivation

The current security scenario of the world has put security related technologies like biometric based person identification into spot-light. Identification of a potential threat from within a crowd or a busy place needs use of non-intrusive biometric and really fast matching and recognition in real time. In such a dynamic security environment, like that of airport security or security in a public gathering, one important constraint on the system is that the face database is always changing and training is a continuous process. In such applications training, that includes calculation of high order matrices of the face vectors in face space and then calculating the Eigen values and Eigen vectors, is done alongside recognition and real time implementation of such computationally intensive training is required.

Most of the pattern recognition applications, like face recognition where some type of decision is made, based on the given data, can be mapped on to a pattern recognition problem. Real-time face recognition does not afford time delay in the algorithm's computation and these applications require time-efficient systems (Kamal 2003, Marwedel 2006). The first step to any pattern recognition solution is separating the meaningful data and this is accomplished through calculation of Eigen values (Yamada 2003).

The accuracy of the Eigen value based system usually depends upon the number of Eigen vectors used for projection and feature extraction. On the other hand, increasing the Eigen vectors means increase in extracted features and increase in time. Therefore, suitable combination of Eigen vectors and decision time could be incorporated for a good face recognition system.

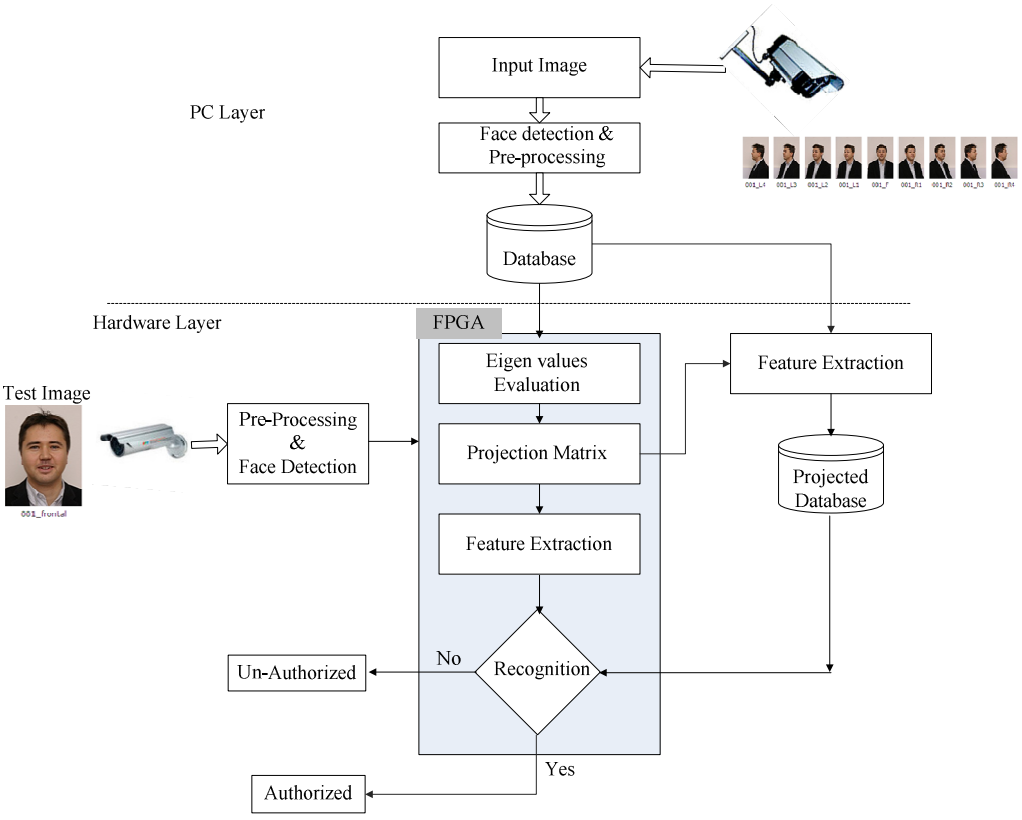


Figure- 1.3 An FPGA-based real-time face recognition system for dynamic environment.

The performance of a real-time system depends upon its accuracy, decision time, power dissipation and storage space. Storage space usually depends upon algorithmic types. Some algorithms generate high dimensional features space but it requires larger space. PCA generates a reduced features space for decision making. However, PCA has higher time complexity and lower accuracy. Fast PCA has reduced the time complexity but it has an instability issue especially for high resolution images. Therefore, some modification in fast PCA may be required to achieve a high accuracy with minimum decision time.

In addition to decision time and storage space, power dissipation is another important issue for a real-time portable face recognition system. Power dissipation depends upon storage space for extracted features and FPGA resources which are required for algorithm realization. Co-design methodology could provide a better combination of execution time and power dissipation on FPGA. Thus, a novel FPGA-based architecture uses an algorithm needing small storage space and it provides decision in linear time complexity with high accuracy using reduced extracted space; thus could be a viable candidate for a real-time face recognition system.

1.6 Objectives

Efficient face recognition systems based on Eigen space and frequency distribution curve (FDC) matching technique have been developed. Hardware realization of these systems provides face recognition in real time. In this thesis, research objectives which have been achieved are:

- Eigen values based efficient face recognition for a dynamic security environment.

- A method has been developed by using Modified Gram-Schmidt Orthogonalization (MGSO) for an efficient Eigen space.
- The implementation of Householder method on FPGA by using co-design pipelined architecture for efficient evaluation of Eigen values has been developed.

- To avoid matrix decomposition and other high order computational matrix operations involved in Eigen values evaluation, a technique based on FDC has been demonstrated for efficient face recognition.
  - FDC is formulated with a bias towards its hardware realization. To improve its performance an adaptive pattern vector controller has been designed for in FDC architecture.
  - FDC based architecture performance was further improved by exploiting hardware parallel processing implementation.

1.7 Thesis Organization

Thesis is organized as follow: Chapter 2 presents a literature survey of face recognition algorithms and their real-time implementation on FPGA. Chapter 3 describes an improvement in face recognition using modified fast PCA algorithm. In chapter 4, FPGA hardware is exploited for parallel execution of non-restoring algorithm. In this chapter, an evaluation of Eigen values for face recognition on FPGA is implemented. Chapter 5 presents efficient evaluation of vector normalization on FPGA. Chapter 6 is about a novel FDC technique for a face recognition on FPGA. Chapter 7 summarizes the contributed work and presents future directions of this research.

CHAPTER 2

FACE RECOGNITION: AN OVERVIEW

2.1 Introduction

The ability to identify or verify the face of a person, using facial characteristics, by a machine is the objective of a face recognition system. In the initial phase of research, algorithms were developed which were inspired by human vision. In the second phase, a face was treated as an entity defined by the combination of features and their mutual relationships. Some algorithms were biased towards local and other towards global features. A hybrid of both had shown better results, however, they were computationally intensive.

Face recognition algorithms can be broadly divided into two categories: a) model-based algorithms and b) appearance-based algorithms. Attempts have been made to develop algorithms related to both categories to achieve accurate results with minimum execution time (Middleton 2006, Cai 2006, He 2005, Moghaddam 1997, Liao 2007, Kim 2002).

For a real-time face recognition system hardware implementation of such an algorithm is preferred. But an efficient hardware realization covers further at least two major milestones. First the algorithm should be developed by considering the hardware chip limitations. Second, hardware architecture should be designed to exploit parallelism to speed up the results. Pipelining and instruction level parallelism are the established hardware parallel processing techniques. But most of the time power dissipation aspect becomes a dominant issue in a high speed system. To get

optimum power dissipation, it is recommended that a portion of the algorithm, which is usually computationally intensive, should be implemented on hardware and the rest in software layer to achieve optimum combination of execution time and power dissipation. Such a co-design architecture could lead to develop a system having low recognition time with minimum power consumption.

2.2 Face Algorithms Evolution

Initially, face recognition algorithms have been treating face as an object, but it is an established fact that the success rate of these algorithms was very low (Tanaka 1993, Kalocsai 1994). The face recognition is different from object recognition because of the following six reasons (Biederman 1997):

(i) There are many behavioral differences called *configurable effects,* which are related to local features of a face like nose or mouth. These features are not part of a non-face object.

(ii) Viewpoint invariant: Different outer boundaries are recognized in case of objects while many faces may have the same outer boundaries. Therefore, the performance of algorithms for objects deteriorates for face recognition.

(iii) Contrast and illumination effects make the face recognition complicated while the simple object recognition is not affected.

(iv) Accessories on the face image like glasses and jewelry affect face recognition but object recognition provides almost same results with accessories.

(v) Rotation in depth also relates only to face recognition.

(vi) Rotation in plane also affects the recognition results of a face image.

By considering the features listed from (i) to (vi), it can be claimed that the face recognition is different from the object recognition and the preceding one is technically more demanding and complex.

In the next phase of algorithms, scientists believed that machine recognition algorithm might work well if it was developed by keeping in mind the psychological process of an infant. Since 1970, psychophysics were assumed the perceptional significance in recognition, which mainly depended on the holistic structure of a face and sometimes perception depends upon a local feature like nose (Kelly 1970). However, perceptional rules would not handle the parameters which affect the facial characteristics (Bruce 1988, Shepherd 1981, Chellapa 1997, Ellis 1986). Later on, experimentally it had been shown that perceptional rules could not completely resolve a face recognition problem (Chellappa 1995) and the researchers then focused on finding the solution by formal methods.

The initial development of formal methods (algorithms) had shown reasonably good recognition results; however, experiments were performed with simple data sets having minor variation in facial expression (Galton 1888). Darwin in 1972 developed an algorithm which performed well with variation in facial expressions and then in 1973 Kanade researched on facial profile (Darwin 1972, Kanade 1977).

When the development of algorithms was in its second phase, the concept of face detection had emerged (Chellapp 1995). First time in mid-1990s, face detection was treated separately when the face was segmented from the background using template matching by employing feature-based and neural network techniques (Chen 1995, Lin

1997, Sung 1997, Rowley 1998). It is worth mentioning that face detection is one of the preprocessing steps towards face recognition (Ming-Hsuan 2002).

2.3 Classification of Algorithms

Broadly, face recognition algorithms can be divided into either appearance-based or model-based algorithms as shown in Figure-2.1. Appearance-based methods are further divided into linear and non-linear transformation techniques (Lu 2003). A linear transformation projects face vectors into basis vectors and coefficient of projections are used to represent image faces (Swets 1996, Belhumeur 1997, Barlett 1998). Whereas, a non-linear transformation represents an image as a data point in a high dimensional face space by considering the geometric structure of an image (Lu 2003). On the other hand, model-based face recognition generates a model based on the prior knowledge of human face (Cootes 2001) and the detail of which is discussed below.

2.3.1 Model-Based Methods

Model-based techniques create a model based on the information gathered during extraction of features (Cootes 1995). Some extraction techniques concern with a single feature of a face e.g., an eye (Hallinan 1991) and the other is concerned with the distance between different features. The extraction of features usually accomplishes using edge detection, structural matching and template-based methods (Kelly 1970, Yuille1992). In template-based methods, every feature requires a specific template (Yuille 1992). For example, a separate template is used for nose while the mouth requires another. This shows that template-based methods have limitations which were addressed using active shape model (ASM) technique (Cootes

1995). Furthermore, to accommodate the texture variations, which were over looked by the conventional model-based techniques, Lanitis in 1995 and cootes in 2001 presented flexible appearance models (FAM) and active appearance model (AAM) respectively (Lanitis 1995, Cootes 2001).

The variation in lighting is another changeling issue of face recognition. Hidden Markov Model (HMM) was developed in 1960s and it was used first for speech recognition in 1970s. In 1990s it was used for face recognition and works well in varying light, facial expression, and orientation of a face image. The HMM uses rows of pixels to compute facial features (Nefian 1998, Samaria 1994, Samaria 1994a) and its prime purpose is to find hidden parameters by using Markov probability process. The combination of image height and overlapping parameters plays an important role in HMM. HMM provides 84% recognition with O_h = 10 and O_L = 9 for ORL face database, where O_h is overlapping blocks and O_L is overlapping height (Kim 2010). However, being probabilistic in nature HMM has high computational cost and may not provide an efficient solution for a real-time system.

Features based graph matching technique using dynamic link architecture (DLA) gives better results than HMM (Okada 1998, Wiskott 1997, Buhmann 1990). Graph based models treat faces as a graph having nodes of facial features like a nose or an eye.

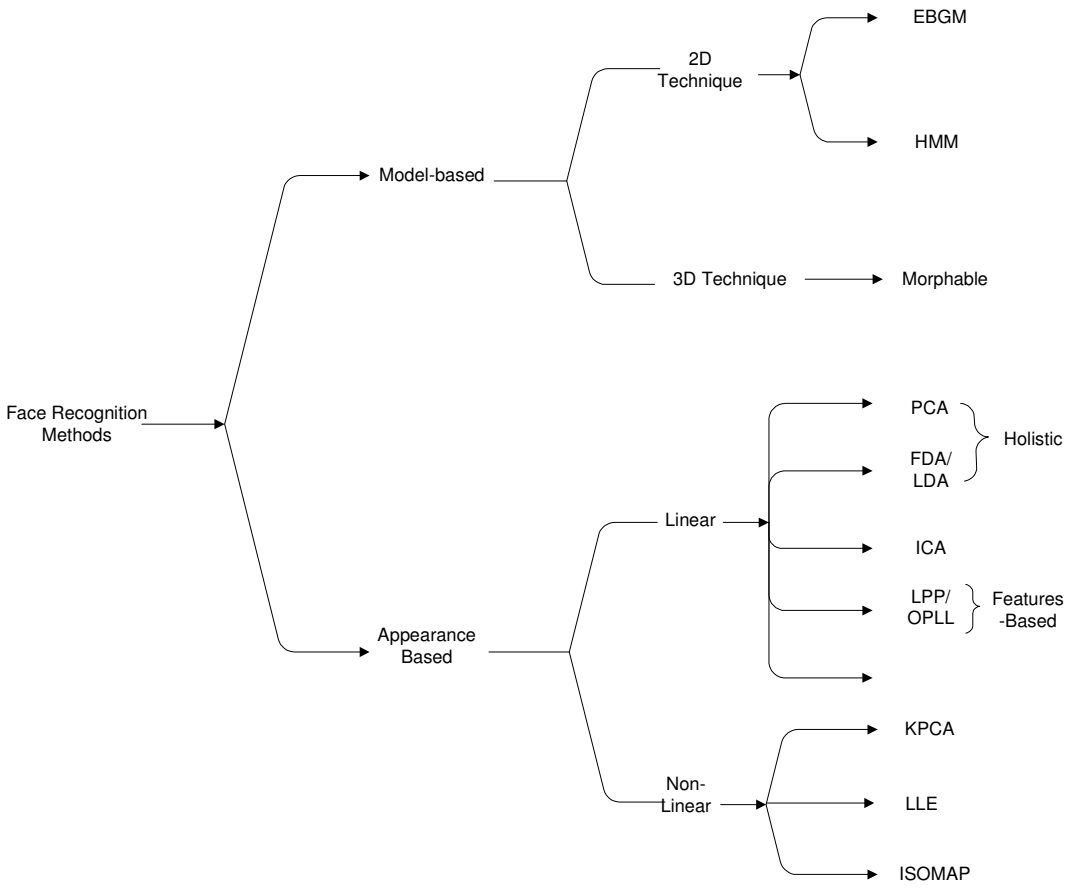


Figure-2.1 Classification tree of face recognition algorithms.

The elastic bunch graph matching (EBGM) technique has shown improvement in face recognition with large size images (Wiskott 1997). EBGM uses DLA for finding neuron's co-relations (Buhmann 1990, Lade 1993). The co-relations of neurons play a significant role in the formation of graph node in EBGM. The DLA also solves the syntactic relationships in neural networks and it generates a dynamic variable that adjusts syntactic weights dynamically (Lade 1993). These weights are significant in signal transmission which helps to generate neuron's co-relations. Improvement has been reported in EBGM model technique using Gabor wavelets for variation in illumination, scaling, distortion and rotation (Manjunath 1992). However, Gabor wavelet is robust for small rotational angle (Manjunath 1992).

Model-based techniques usually compute node and edge values of a graph using complex wavelet co-efficient. Another earlier effort in model-based technique is the area of three-dimensional (3D) modeling of a face (Gordon 1991). The 3D face model can be constructed using three face images of a person by morphable technique (Blanz 1999). A 90% success rate of 3D morphable technique for six subjects is reported by (Lu 2003, Zhang 2005).

2.3.2 Appearance Based Methods

Significant improvement in face recognition systems has been observed using appearance based methods (Rowley 1998, Sung 1997). Appearance-based methods are divided into linear and non-linear mapping techniques.

Linear Mapping:

Linear mapping by definition reduces the image space to an efficient face space. PCA, independent component analysis (ICA), linear discriminant analysis (LDA), locally linear embedding (LLE), locality preserving projection (LPP) and orthogonal locality preserving projection (OLPP) are the main linear mapping techniques for face recognition (Lu 2003). PCA and FDA or LDA are holistic in nature while LPP and OLPP also consider the local details of an image (He 2005). PCA and Karhunen-Loeve (KL) have the greatest strength of vector space reduction and interpretation (Kirby 1990, Sirovich 1987). In pattern recognition applications PCA is extensively used for the reduction of the computational space. In PCA, Eigen faces are first computed then each face is represented by a vector of specific weights. PCA has a minimum time complexity $O(n^3)$ for computing basis vectors. Eigen faces use PCA to compute first and second order dependence when distribution of input information is Gaussian. The Eigen approach was extended to Bayesian for the measurement of similarities amongst images, but Bayesian is stochastic and thus computationally intensive than PCA (Moghaddam 1997). Using PCA many image processing techniques have been developed and they are discussed in (Li 1999, Belhumeur 1997, Liu 2001, Swets 1996, Zhao 1998, Phillips 1998, Moghaddam 1997).

ICA can extract independent components from dependent ones without prior knowledge of correlated elements like blind source separation (BSS) (Cardoso 1998). ICA computes first and second order dependency regardless of input distribution using Kurtosis maximization process (Daugman 1989, Field 1994, Bell 1997, Lewicki 2002). By involving higher order statistics, ICA provides better results than PCA when trained with a bigger data set and it maximizes the information transfer in

a more general form (Bartlett 1998, Draper 2003). However, ICA is a generalized form of holistic algorithm and it cannot display local properties perfectly especially for the local salient features having pixels with non-zero values. Although, ICA provides linear mapping yet its computation cost grows significantly with increasing image resolution (Dagher 2006). Furthermore, ICA axes are not orthogonal, thus it offers poor discriminating power within classes (Hu 2007).

Fisherfaces usually performs better than ICA when the data of a class lies near a linear subspace and a small number of classes are involved (Hu 2007, Lu 2003). Fisherfaces uses LDA between classes/subjects (Hu 2007). LDA computes variance between and within the classes using original sample space (He 2005, Eleyan 2006). Projections between the classes can be deduced from within the class variance multiplied with corresponding Eigen values (Belhumeur 1997). Therefore, LDA has more time complexity and uses supervised learning, which is not advisable for a real time automatic system.

LPP is an appearance based method used in face recognition which preserves local features using optimal linear function (He 2005). LPP uses approximate Eigen function to construct a manifold structure of a face. The face manifold structure is developed by using nearest-neighbor graph technique (He 2005). LPP generates low dimensional space for the evaluation of nearest-neighbor graph using objective function having normalization of vectors. Normalization of vectors on iteration makes the LPP a computationally intensive algorithm than PCA and LDA.

On the other hand, LPP provides non-orthogonal basis function and such function may not reconstruct the original data accurately. Whereas, orthogonal LPP

constructed by adjacency graph provides better recognition than LPP (Cai 2006). A weight matrix is calculated by using adjacency graph to construct a Laplacian matrix which provides orthogonal space. However, computational cost of these higher order matrices makes the system slower than LPP.

A recent work by Kussul based on permutation coding claims 99.9% recognition accuracy, but the technique was checked only for ORL face database which is not sufficient to demonstrate the validity of the concept (Kussul 2006). Another recent work reported 81% recognition efficiency by using one template matching and it was also tested on ORL database (Huang 2009).

To improve the recognition time of a real-time system, low time complexity of an algorithm is one of the essential ingredients. Fast PCA improves the time complexity of PCA from $O(n^3)$ to $O(n^2)$ and also gives control to select number of leading Eigen vectors (Sharma 2007). Fast PCA uses Gram-Schmidt orthogonalization to evaluate Eigen vectors for a co-variance matrix. On the other hand, co-variance of an image having high resolution may affect the convergence behavior of Fast PCA (Sajid 2009). Therefore, modification in Fast PCA could be useful task for an efficient real-time system.

Non-Linear Mapping:

The results are not impressive by linear mapping when general data sets are used because, these techniques calculate Euclidean structure in the face space, which fails to provide a successful face recognition if the images lie on a nonlinear sub manifold face space (Martinez 2001). The human face is a manifold entity in general and can be better treated by a non-linear mapping (Chang 2003). A lot of efforts have been made

to investigate the nonlinear manifold structure of a face (Roweis 2000, Seung 2000, Chang 2003). Kernel PCA (KPCA) and isometric mapping (ISOMAP) are the two famous nonlinear techniques which have shown relatively better results than existing linear techniques on specific crafted data sets (Belkin 2001, Tenenbaum 2000, Roweis 2000, Kim 2002, Saul 2003).

Non-linear mapping produces a big computational space even for a medium level input parameters. The recognition is then performed by using linear support vector machine (SVM) technique (Scholkopf 1996, Kim 2002).

ISOMAP and KPCA work similarly to reduce high dimensional space in non-linear transformation. KPCA computes principal components in high dimensional feature space which relates to non-linear input variables. ISOMAP uses geodesic distances between data points and then uses Multi-Dimensional scaling to induce low dimensional manifolds (Yang 2003). Since ISOMAP reconstructs basis vectors using geodesic metric, it would not provide optimal results for the classification of data points.

Recent non-linear second order of modular transformation using LLE method was first introduced in 2000 (Roweis 2000). LLE enables preservation of local details of K-nearest neighbors, which best describes each data point within its vicinity using linear coefficients of transformation (Saul 2003). The technique shows good results when samples are large in number whereas, in a real-life, face recognition application, only few samples are available for an individual.

In general, nonlinear transformations are computationally intensive compared to linear transformation commonly known as low dimensional transformation. Low

dimensional transformation has longest time era in research history as shown Table-2.1. The table presents timeline, research contributors and their corresponding techniques pertaining to face recognition. Linear transformation offers small feature space compared to input space. Contrary to linear transformation, non-linear transformation manifolds the input space as defined by Mercer's theorem, i.e., $kk(s_j, s_y) = \varphi(j).\varphi(y)$ (Lu 2003).

2.4 Hardware Based Face Recognition Systems

It is a well established fact that a dedicated hardware based implementation of an algorithm is faster than its software based execution. Microsoft presented a processor based system which can handle two frames per second, when implemented on a 700 MHz Intel or a 200 MIPS-strong ARM processor (Viola 2004). This recognition rate is considered slower for real world applications. Another hardware-based face recognition system using a artificial neural networks and Eigen faces is implemented on an analog ADSP-BF535 EZ-KIT device (Wei 2004). This system provides recognition with a maximum accuracy of 80%, consumes 36 m sec, and uses more than one MB of storage for each face. A multi-processor architecture is developed by Fatemi that included a smart camera and gave recognition in 4.3 m sec with 90% accuracy (Fatemi 2003).

Table-2.1: Timeline chart of research development for face recognition in various categories.

Timeline	Major Contributors	Major Techniques	Major category of techniques
1981- 1986	Bruce, Kelly, Shepherd and Ellis	Perceptional rules for facial characteristics	Structure-based
1993-1997	Biederman, Tanaka and Kalocsai	Neurocomputational basis vectors and differentiate objects from faces.	Structure-based
2000-2003	Roweis, Seung and ChangLu, Scholkopf, Kim, Yang	Nonlinear manifold structure, linear support vector machine, ISOMAP, Locally Linear Embedding, KPCA	Appearance-based (non-linear transformation)
1992-2005	Buhmann, Blanz,Okada, Nefian, Gordon, Manjunath, Lu, Lade , Zhang and Wiskott	Hidden Markov Model, Flexible appearance models, active appearance model, Dynamic Link Architecture, Improvement in EBGM using Gabor wavelets, features based graph matching, elastic bunch graph matching (EBGM) and 3D modeling of a face	Model-based
1989-2009	Belhumeur, Barlett, Li, Liu, Swets, Zhao, Phillips, Moghaddam, Eleyan, Huang, Sharma,	PCA, FAST PCA, independent component analysis (ICA), linear discriminant analysis (LDA), locally linear embedding (LLE), locality preserving projection (LPP), orthogonal locality preserving projection (OLPP).	Appearance-based (linear transformation)

Co-design methodology is considered a better technique for a real-time face recognition system. This could provide an optimal partitioning of an algorithm (Saha 2007, Zaki 2004) for its hardware and software based implementation.

The technique can provide an optimum combination of time and power parameters without compromising the accuracy of the algorithm. Such an implementation is expected to recognize faces on real time basis as required by an application and can be used in portable devices.

2.5 Summary

Model-based techniques inherit approximations in generation of graphs due to the prior information of the human face. Model-based methods are more error prone than appearance based methods. Linear holistic methods are suitable for efficient face recognition compared to feature-based methods, because they are able to reduce a computationally intensive image space to a reduced face space. But they need large number of sample images per subject for reasonable accuracy. Hybrid linear techniques again become computationally intensive since they generate feature space enriched with local and holistic information. Thus, an linear algorithm which could provide accurate face recognition in linear time scale using minimum sample images per subject could be a suitable candidate for a real-time system.

CHAPTER 3

MODIFIED FAST PCA

3.1 Introduction

PCA is one of the basic holistic algorithms which has been used in many pattern classification applications, such as in face recognition and object identification (Martinez 2001, Xiaofei 2005, Zhao 2003, Chellappa 1995, Daugman 1997, Pankanti 2000, Pentland 2000). PCA may also be used as a preprocessing step in other holistic techniques like fisher, linear discriminator, locality preserving projection (LPP) and orthogonal LPP (Zhao 2007). Its prime focus is to extract a low-dimensional feature space called as subspace from the original feature space by finding a suitable projection matrix. The projection matrix is made up of basis vectors of the subspace which are actually the Eigen vectors corresponding to largest Eigen values of the co-variance matrix. While doing the dimensionality reduction, it tries to keep intact most of the original information of the data (Turk 1991). Eigen values are commonly calculated by employing the decomposition method (Sharma 2007). However, fast PCA, which is a variant of original PCA, uses Gram-Schmidt orgothonalization (GSO) method to evaluate Eigen values. It is worth mentioning that GSO based Eigen values calculation technique uses orthogonalization in low-dimensional space which is time-efficient because, it reduces complexity of decomposition from $O\ (n^3)$ to $O\ (n^2)$ (Sharma 2007, Sajid 2008-b, Sajid 2009-a).

It has been demonstrated by many researchers that the use of orthogonalization for the extraction of feature vectors in face space produces better results compared to decomposition (Krisnamoorthy 2007-a, Krisnamoorthy 2007-b, Haifeng 2007)

because, it has better discriminating power. However, orthogonalization in GSO is sensitive towards rounding effects due to finite precision of a machine which may cause instability in fast PCA (Burden 1997, Giraud 2003, Sharma 2007, Sajid 2008-b, Sajid 2009-a).

It has been demonstrated by numerous numerical experiments that GSO may produce a set of non-orthogonal vectors which may lead to a non-convergence state of fast PCA (Bjorck 1967, Bjorck 1996, Luc 2005). In modified Gram-Schmidt orthonormalization (MGSO) vectors are normalized prior to the orthogonalization process which minimizes the generation of non-orthogonal vectors. This property of MGSO could be used in fast PCA to overcome its instability for the evaluation of Eigen values.

For a time-efficient decision support system, the selection of leading Eigen vectors (LEVs), which are generated by MGSO, is a critical task. Fast PCA has the capability to compute efficiently the desired number of LEVs, whereas decomposition continues till the diagonal of a decompose matrix is filled. Proper selection of LEVs plays an important role in maximizing the variance (Xiaofei 2005, Turk 1991, Simon 2008). It is reported that the variance defined by three LEVs, usually occupies more than 90% of the total space (Siwabessy 1999). Therefore, further inclusion of LEVs may not give considerable improvement in a face recognition system. However, the selection of optimum number of LEVs is not a trivial decision (Rzempoluch 1998, Liu 2000) and is dependent upon sample space and system reliability.

It is an established fact that the first LEV occupies maximum portion of variance and the second LEV shares the second highest and so on (Rzempoluch 1998). In general,

the recognition accuracy in Eigen space is dependent upon number of LEVs used in reduced subspace (Simon 2008). However, the system performance starts saturating after a certain number of LEVs (Martinez 2001, Xiaofei 2005, Turk 1991). On the other hand, recognition/decision time of a system increases by increasing the number of LEVs. Thus, improvement in a system accuracy, within the permitted decision time, requires an optimum selection of LEVs.

The accuracy of a face recognition system, apart from the optimum number of LEVs, also depends upon threshold value used to generate ROC curves. ROC curve is a graphical representation of FAR and FRR, which is used commonly to assess the performance of a face recognition system (Zhao 2003, Toole 2007, Yan 2007). To minimize the system error and to enhance its accuracy, it is imperative to know minimum error value on ROC for a given number of LEVs.

In this chapter, fast PCA has been modified by incorporating MGSO to improve its stability and accuracy for a face recognition system. A technique has been developed to evaluate optimum number of LEVs by using modified fast PCA. Further, a mechanism is suggested to achieve the best combination of system variables for its improved reliability and accuracy.

3.2 Principal Component Analysis (PCA)

Let there be *d* training images, which are represented by the matrices $I_1, I_2,.....,I_d$ of order $r \times c$ each. To mild the lighting effects, each of these matrices is processed using $I_i' = \frac{(I_i - m_i) \times ds_i}{s_i} + dm_i$ where $i = 1, 2,....., d$, m_i is the mean, s_i is the standard deviation, dm_i is the desired mean and ds_i is the desired standard deviation (Mittal

2008). After preprocessing, matrices are converted into column vectors $J_1, J_2, ..., J_d$ each of length *rc*. Let $\bar{J} = \frac{1}{d}\sum_{i=1}^{d} J_i$ and $A_{rc\times d} = [A_1 \quad A_2 \quad A_3 .... \quad A_d]$, where $A_i = J_i - \bar{J}$ for $i = 1, 2, ..., d$. Then the covariance matrix is $C_{rc\times rc} = AA^T$. To find the significant Eigen vectors of *C*, $L_{d\times d}$ is constructed as: $L = A^T A$ where $d << rc$ (Turk 1991). Suppose $\lambda_1, \lambda_2, ..., \lambda_d$ are the Eigen values (in descending order) of *L* corresponding to the Eigen vectors $u_1, u_2, ..., u_d$ respectively. The desired LEVs are then computed by employing the decomposition of square symmetric matrix *L*.

3.2.1 Householder Method

Numerically computed Eigen values by decomposition methods are more accurate than orthogonal methods but computationally intensive (Sharma 2007). Decomposition is considered a superior method as compared to Jacobi transformation for symmetric matrices (Press 1992, Golub 1992). There are many decomposition methods out of those Householder, (HH) is considered an efficient one for Eigen solutions because it terminates definitely after *n*-2 iterations, where *n* is the rank of an input symmetric matrix (Press 1992). The objective of Householder transformation is to reduce first a real symmetric matrix into tridiagonalization form. Then, tridiagonalization of a given symmetric $A_{rc\times d}$ matrix into Householder orthogonal matrix *P* requires a real vector *w* for the following computation:

Let $P = 1-2w.w^T$ where $w.w^T$ is outer product and $w^T.w$ is inner product. Then

$P^2 = (1-2w.w^T).(1-2w.w^T)$

$P^2 = 1- 4w.w^T + 4w(w^T.w)w^T$

$P^2 = I$

It means $P=P^{-1}$, but $P^T=P$, implies $P^T= P^{-1}$ proving orthogonality.

It can be written $p = 1 - \frac{u.u^T}{H}$ where H is scalar, u is any vector.

$H \equiv \frac{1}{2}|u|^2$ …… Eq (3.1)

Let x be a vector made up of first column of A. Then $u = x \mp |x|e_1$ be a vector, having e_1 as a unit vector $[1,0,\cdots,0]^T$. Then it is easy to prove that $P = \mp|x|e_1$ .This shows that in HH a matrix P acts on a given vector x and converts its elements to zeros except the first one.

To reduce $A_{n\times n}$ into the tridiagonal form,

let $A' = P.A.P,$ where $P = 1 - \frac{u\,u^T}{H}$.

Then, $A' = \frac{A.u}{H} . A . (1 - \frac{u\,u^T}{H}) = A - pu^T - up^T + 2Ku.u^T$

where $K = \frac{u^T.P}{2H}$ is a scalar and $p = \frac{A.u}{H}$.

Defining $q = p - Ku$, we have $A' = A - q.u^T - u.q^T$. …… Eq (3.2)

At m^{th} stage ($m=1,2,\ldots,n-2$), the vector u is written as

$u^T = \left[a_{i1}, a_{i2,}a_{i3}, \cdots, a_{i,i-2}, a_{i,i-1} \mp \sqrt{\sigma}\,, 0, \cdots, 0\right]$,

where $i = n - m + 1 = n, n-1, \cdots, 3$ and $\sigma = (a_{i1})^2 + (a_{i2})^2 + \left(a_{i,i-1}\right)^2$.

Once $\sigma, u, H, P, K, q, A'$ are computed, Eigen vectors of A can be computed by applying the transformation $Q = P_1.P_1.\cdots.P_{n-2}$.

Once all the P have been computed, the matrix Q is obtained by following recursive procedure:

$$Q_{n-2} = P_{n-2}$$

$$Q_{n-3} = P_{n-3}Q_{n-2}$$

$$\vdots$$

$$Q = Q_1.$$

Once a real symmetric matrix has been reduced into a tridiagonal form, its Eigen values can be found by Newton's method. However, decomposition technique is more efficient than Newton's method (Press 1992). Decomposition technique can be classified as: a) lower triangular decomposition as: b) upper triangular decomposition. The lower triangular decomposition of a matrix has smaller round-off affect than upper triangular matrix decomposition (Burden 1997). Therefore, lower triangular decomposition has been implemented in proposed architectures.

3.3 Fast PCA

Fast PCA has been proposed using GSO for the evaluation of Eigen values in $O(n^2)$ time where n is rank of co-variance matrix. The leading Eigen vector has been evaluated first and subsequently (h-1) Eigen vectors will be computed in a descending order (Sharma 2007).

The algorithm computes leading Eigen values uses GSO.

Step-1: h represents the number of leading Eigen vectors.

Step-2: Compute the co-variance matrix $C = A^T A$ *and define a scalar* $p \leftarrow 1$.

Step-3: To populate an Eigen vector Φ_{pp} *with randomly generated values between 0 and 1.*

Step-4: Gram-Schmidt orthogonal process computes basis vectors as:

$$\Phi_{pp} \leftarrow \Phi_{pp} - \sum_{j=i}^{pp-1} \left(\Phi_{pp}^T \Phi_j\right) \Phi_j$$

Step-5: Normalize the vector computed in previous step using norm2

Step-6: Go to step 3 in case $abs(\Phi_{pp}^{+T}\Phi_{pp} - 1) < \varepsilon$ *is failed.*

Step-7: Increment the value by one of p and go to step 2 until p = h.

3.4 Limitation of Fast PCA in Face Recognition

It has been shown in (Sharma 2007) that both PCA and fast PCA generate similar MSE and converge in finite iteration with the following condition:

a) Epsilon or threshold value for the convergence condition is 0.01.
b) Features vectors (*d)* of orthogonal space should be equal to 100.
c) The value of *h* (principal components) should be 10.
d) Matrix element should be less than 4000 pixels.

By considering the aforementioned conditions fast PCA is used for face recognition. It has been observed that a fast PCA based face recognition system gives convergence for resolution 118 x 112 whereas, it fails to converge for high resolution images. Table-3.1 shows the non-convergence of fast PCA in face recognition for *d*=100 samples images having pixels 3894. Non-convergence in fast PCA is addressed when iterations in GS process exceed 1000. These non-convergence conditions are highlighted with shaded cells in Table-3.1.

The convergence criterion in fast PCA is based on the rule that dot products of orthogonal vectors, which are generated by GS process, should be unity (Anton 2005). The products are calculated with current ϕ_{pp}^{+T} and previous ϕ_p base vectors and its difference from unity is computed as

$$(\phi_{pp}^{+T}\phi_p - 1) \qquad \text{…… Eq (3.3)}$$

For convergence condition, the value of Equation (3.3) should be less than the chosen value ε . It is observed that when variation in poses is less than six in the TS having

less than 100 images then Equation (3.3) approaches slowly to ε. This slow convergence is because of the stochastic and iterative nature of GS process.

Table-3.1 Limitation of Fast PCA when $\epsilon = 0.01$, varying dimensionalities whereas shaded cells show non-convergence conditions.

Image Size	*h*	Faces / Object	Single / Multi pose	Max. Iteration	Training Images
66 x 59	10	Faces	Multi	10	all
66 x 59	10	Faces	Multi	> 1000	10
66 x 59	10	Faces	Single	> 1000	10,20
66 x 59	10	Faces	Three	> 1000	10
66 x 59	10	Object	Single	11	10,20
118x 122	10	Faces	Muti	10	All
118x 122	10	Faces	Single	> 1000	10,20
118x 122	10	Faces	Three	> 1000	10,20
118x 122	10	Object	Single	> 1000	all

In case of non-convergence, GS multiplies the co-variance matrix with the modified vector ϕ_{pp} and then checks the convergence condition.

Whereas the initial values of ϕ_{pp} depend on the system random generation process. GS process tries to orthogonalize the current ϕ_{pp}^{+T} and previous ϕ_{pp} vectors until it is less than ε which defines tolerance of the system. Ideally this tolerance should be zero, but practically the value of ε varies according to the randomness of ϕ_{pp} in fast PCA and rounding effect in all sorts of other iterative processes (Burden 1997).

3.5 Modification in Fast PCA

The Fast PCA has been used GSO for the evcaluation of orthogonal basis vectors. However, Fast PCA is not suitable for face recognition because of unstability in the convergence step of the algorithm. Modified GSO is introduced for the stable evaluation of basis vectors for face recognition images having high resolution. The algorithm with MGSO is stated as:

In this algorithm, the iteration counter is denoted by *IC,* the maximum number of allowed iterations by *M and* the number of non-zero columns of a matrix *U* by $\beta(U)$.

Step (i) *Initialize a zero matrix U of the order of* $d \times h$. *Set IC* = 0*, assign a positive integral value to M. Choose a sufficiently small tolerance* $\varepsilon > 0$ *e.g. (*10^{-2}*).*

Step (ii) *Update U as suggested in step (viii) (ignore this step while calculating the 1st leading Eigen vector).*

Step (iii) *Randomly select a vector* ψ_o *of size d.*

Step (vi) *Normalization of* ψ_o *is computed as* $\psi_1 = \frac{\psi_o}{\|\psi_o\|}$ where $\|\psi_o\|$ is Euclidian length norm of a vector.

Step (v) *Compute* $\psi_2 = L\psi_1$.

Step (vi) *For calculating the first leading Eigen vector, set* $\psi_3 = \psi_2$, *otherwise*

$$\psi_3 = \psi_2 - \left(\sum_{j=1}^{\beta(U)} \left(\psi_2^T u_j \right) u_j \right)$$ *where* u_j *is the* j^{th} *column of U.*

Step (vii) *Normalization of* Ψ_2, Ψ_3 *are computed as* $\Psi_4 = \frac{\Psi_3}{\|\Psi_3\|}, \Psi_2 = \frac{\Psi_2}{\|\Psi_2\|}$ *where* $\|\Psi_2\|$ *and* $\|\Psi_3\|$ *are Euclidian length norms of* Ψ_2 *and* Ψ_3 *respectively.*

Step (viii)

(a) If $\left| \psi_4^T \psi_2 - 1 \right| < \varepsilon$, , *go back to step (ii) and replace the 1st zero column of U by* ψ_4.*Then If there is a zero column in U repeat step (iii) to (viii) with a different choice of* ψ_o *in step (iii), otherwise go to step (x)*

(b) If $\left| \psi_4^T \psi_2 - 1 \right| \geq \varepsilon$, *go to step (ix).*

Step (ix) *Update IC by IC=IC+1. if IC < M, then repeat step (v) to step (viii) with* $\psi_1 = \psi_4$ *in step (v), otherwise go back to step (iii).*

Step (x) *Compute Eigen values of L* by $U^T L U$.

As discussed in (Turk 1991), Au_j are the Eigen vectors of *C*. Now define

$$t_j = \frac{Au_j}{\|Au_j\|} \text{ for } j = 1, 2, 3, ..., h \text{ and } T_{rcxh} = [t_1\ t_2\ .....t_h].$$

Then collect the projection of each of the *h* leading Eigen vectors on A_i in a matrix $W_{d\times h}$ as:

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

Let N be a column vector of length *rc,* representing an input image. Define three column vectors D_N, W_N, V_N, and a number *r* as:

$$D_N = N - \bar{J} \quad \text{..... Eq (3.4)}$$

$$W_N(j) = (D_N)^T t_j; \quad j = 1, 2, ..., h \quad \text{..... Eq (3.5)}$$

$$V_N(i) = \left\| W_N - (i^{th} column of\ W^T) \right\|; \quad i = 1, 2, 3, ..., d. \quad \text{..... Eq (3.6)}$$

$$r = \max(V_N) - \min(V_N). \quad \text{..... Eq (3.7)}$$

Suppose there are *k* input images $N_1, N_2, ... N_k$ and half of which are unknown. Each image is represented by a column vector of length *rc*. Repeat (3.4) to (3.5) for each $N_1, N_2, ... N_k$ image to get

$r_1, r_2, ... r_k$ and let $R = [r_1\ r_2 ..... r_k]$. Calculate the mean of *R* and denote it by *Z*. Define

$$Z_p = pZ, \text{ where } p = 0, h', 2h', ..., 2 \text{ for some } h' > 0. \quad \text{..... Eq (3.8)}$$

For each *p*, define a row matrix H_P of length *k*, by

$$H_p(i) = \begin{cases} 1 & \text{if } r_i < Z_p \\ 0 & \text{otherwise} \end{cases}$$

where $i = 1, 2, 3, ..., k$.

Associated with each *p*, define two numbers F_{RP} & F_{AP} for any of the following two cases:

Case-I: When *k* is even,

F_{RP} = Number of zeros in *Hp(i)*; $i = 1, 2, ..., \frac{k}{2}$

F_{AP} = Number of ones in *Hp(i)*; $i = \frac{k}{2} + 1, \frac{k}{2} + 2, ..., k$

Case-II: When *k* is odd,

F_{RP} = Number of zeros in *Hp(i)*; $i = 1, 2, ..., \frac{k-1}{2}$

F_{AP} = Number of ones in *Hp(i)*; $i = \frac{k+1}{2}, \frac{k+1}{2}+1, ..., k$.

For each *p*, define a false rejection rate (FRR_p) and a false acceptance rate (FAR_p) as follows:

$$FRR_p = \begin{cases} \frac{2F_{RP}}{k} \times 100, & if\ k\ is\ even \\ \frac{2F_{RP}}{(k-1)} \times 100, & if\ k\ is\ odd \end{cases}$$

$$FAR_p = \begin{cases} \frac{2F_{AP}}{k} \times 100, & if\ k\ is\ even \\ \frac{2F_{AP}}{(k+1)} \times 100, & if\ k\ is\ odd \end{cases} \qquad \text{…… Eq (3.9)}$$

Denote the sum of the above two by S_p, i.e.,

$$S_p = FRR_p + FAR_p. \qquad \text{..... Eq (3.10)}$$

For obvious reasons, accuracy of results depends upon *h* and *p*. We will now implement our scheme for different choices of *h* and *p* to select their optimum combination to achieve the required accuracy for a given time.

3.5.1 Optimization in Modified Fast PCA

Let us assign different values to $h_i = h_1, h_2, ..., h_q$. To achieve global minima, S_p has been simulated by using least square method. Denote the minimum value of Q_i by: $\min(Q_i); i = 1, 2, ...q$ and the value of *p* for which Q_i attains its minimum value by $p_i; i = 1, 2, ...q$. Suppose $\tau_i; i = 1, 2, ...q$, takes time in decision making when *h* takes values $h_i; i = 1, 2, ...q$. Now write a matrix $F_{q\times 4}$ as follow:

$$F = [H \quad P \quad \min(Q) \quad \tau] \qquad \ldots\ldots \text{Eq (3.11)}$$

Where,

$$H = [h_1\ h_2 \ldots\ldots h_q]^T, \quad P = [p_1\ p_2 \ldots\ldots p_q]^T, \ \tau = [\tau_1\ \tau_2 \ldots\ldots \tau_q]^T \text{ and},$$
$$\min(Q) = [\min(Q_1)\min(Q_2)\ldots\ldots\min(Q_q)]^T$$

The following methodology provides optimum selection of *h* and *p* for a given time and accuracy constraints.

***(a)-Case-I**:*

When the system is required to reach a decision in a time not exceeding τ^*, interchange different rows of *F* in such a way that the 4th column of *F* is converted into ascending order. Denote the resulting matrix by G_1. Let the first *m* entries of the 4th column of G_1 not exceed τ^*.Find the smallest of the first *m* entries of the 3rd column of G_1. Let this be the n^{th} one $(n \le m)$. Then the n^{th} entries of the 1st and the 2nd columns of G_1 are the optimum choices of *h* and *p* respectively. If all the entries of the 4th column of G_1 are greater than τ^*, then the system cannot meet the requirements. In this case, the first entry of the first and the second columns of G_1 is the best choice of *h* and *p* respectively.

***(b)-Case-II**:*

When the system has to make a decision subject to a maximum error Q^*, interchange the rows of *F* so that the 3rd column of *F* is converted into the ascending order. Let the new matrix be denoted by G_2 and also the first *n* entries of the third column of G_2 are $\le Q^*$.Let m^{th} entry of the 4th column of G_2 be the smallest of the first n^{th} entries of this column. In this case the m^{th} entries of the 1st

and the 2nd column of G_2 represent optimum combination of *h* and *p* respectively. If none of the entries of the 3rd column of G_2 is less than or equal to Q^*, then the first entry of the first and second columns of G_2 is the best choice of *h* and *p* respectively.

***(c)-Case-III**:*

Consider the case when the system is made to take a decision by comparing simultaneously the constraints discussed in (a) and (b). First apply (a) and denote the resulted values of *h* and *p* by h_a and p_a respectively and then apply (b) to get h_b and p_b. Denote the average of h_a and h_b by $\overline{h}$ and that of p_a and p_b by $\overline{p}$ .The average $\overline{h}$ and $\overline{p}$ represent optimum values of *h* and *p*.

3.5.2 Modified Fast PCA Based System Architecture

Architecture and data flow diagram of the proposed system illustrates storage and flow of data amongst various components of the system as shown in Figure-3.1. Architecture is divided into three layers. Covariance matrix and the desired number of LEVs are provided as inputs to MGSO module in the learning layer of the architecture to evaluate Eigen values. Eigen faces are then computed after extracting the feature vectors which are then stored in a DB.

In an optimization layer, an adaptive classifier provides an initial *p* to the threshold balancer on the basis of Eigen faces and sample test images. After getting a new *p* from the threshold balancer, the adaptive classifier defines FAR, FRR, decision time and corresponding number of LEVs, which are subsequently stored in a database for simulation purposes.

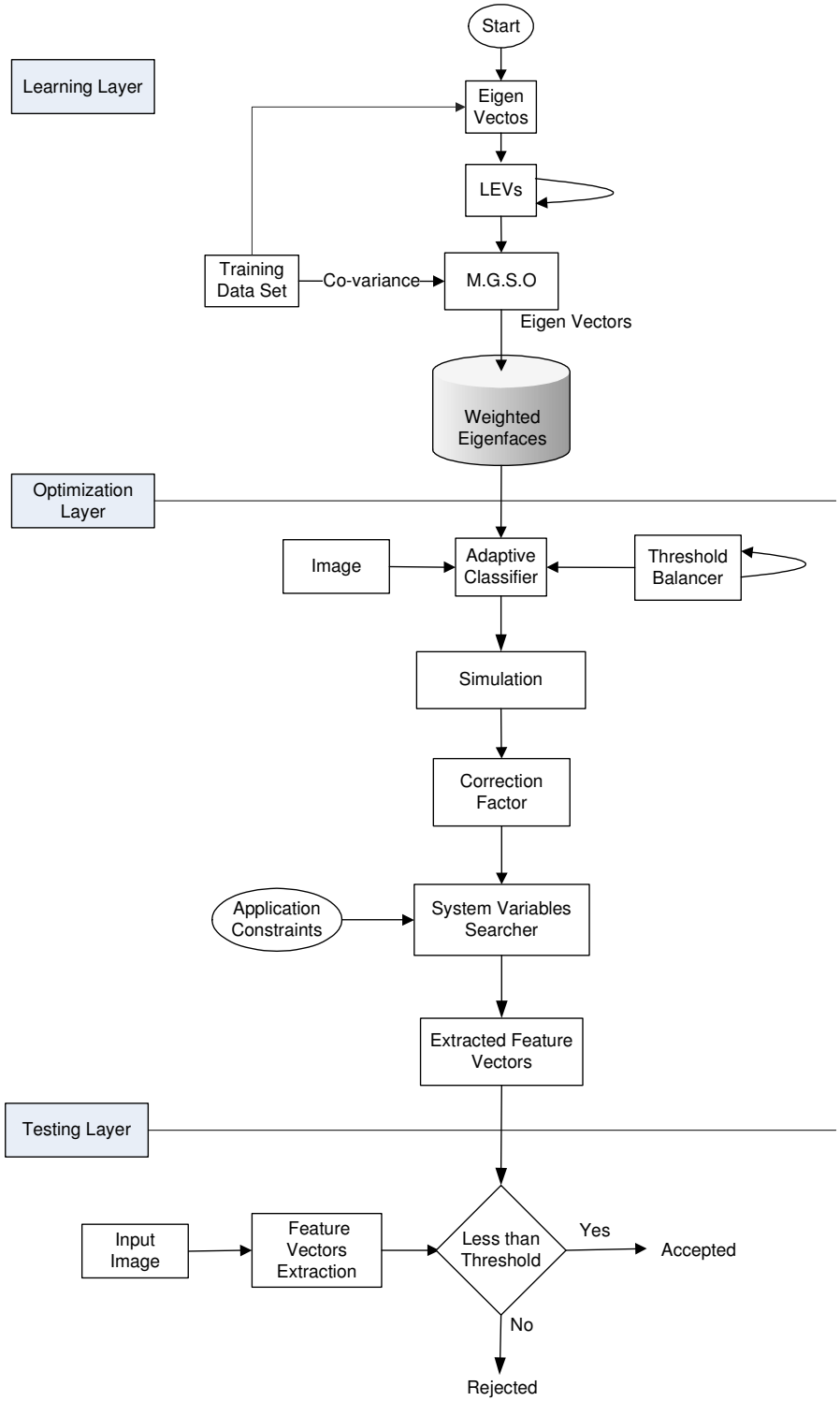


Figure-3.2 A three layer architecture of face recognition based on modified Gram-Schmidt orthogonalization for optimum combination of decision time, minimum error rate and leading eigenvectors.

The sum of FAR and FRR (S_p) is plotted against p and the same is simulated by the least square fitting module to generate convex functions. Whereas, the minimizer provides minimum error by applying LSA on the generated convex functions.

Maximum allowable time for recognition and/or minimum acceptable accuracy of a system is referred to as application constraints. On the basis of these constrains and by using a correction factor, the optimizer of Figure-3.1 provides best available h and p for a given application. Finally, in the testing layer on the basis of optimized h and p, features vectors are extracted and stored in DB. On the other hand, feature vectors of an input image are also extracted and compared with stored feature vectors. This enables the system to accept/reject an input image.

3.6 Face Recognition with Modified Fast PCA

The system is developed as shown in Figure-3.1. To perform different experiments of the proposed system and compare the results with the classical decomposition-based method requires various training sets. The data sets were designed of male and female having images of 270 x 300, 180 x 200 and 90 x 100 resolutions (Spacek 2008). There were small changes in the face position because these images have been acquired in speech mode with no variation in hair style.

Variation in lighting was also there. A total of 120 subjects with 20 images per subject having gray background with minor variation in head turn, tilt and slant were present. Out of 2400 images, 6 TSs 200, 160, 130, 100, 70, and 50 images were made.

3.7 Decomposition versus Modified Fast PCA

It has been observed decomposition-based and modified Fast PCA systems response regarding accuracy, learning and decision time varies by varying the number of training images, their resolution and images per subject. To reveal this complexity, three testing sets namely TS_H, TS_M and TS_L having image resolutions 270 x 300, 180 x 200 and 90 x 100 respectively are made for time efficiency aspect. These testing sets relate to six training sets having properties as shown in Table 3.2. This table shows combination of samples per subject and number of leading Eigen values used for extracted features space.

The time efficiency aspect has been demonstrated using 138 experiments by for the modified Fast PCA and 18 by using PCA decomposition based technique for comparison purposes. When the system does not recognize an image which belongs to the TS at testing phase, it is treated as a true error. And if the system gives positive decision for an image which does not relate to training subjects then it is noted as a false error. The time required by the system for the extraction of features vector is known as learning time while the time required for testing of 20 images is treated as decision time. Sum of these two time slots is considered as the total time required by the system.

Figure-3.2 (a), (b) &(c) illustrates error rate which is the sum of true and false errors, decision and learning time of a PCA based system for the data set having resolution of images 270 x 300, 180 x 200 and 90 x 100 respectively. It is observed that the decision and the learning time both increases by increasing with the number of leading Eigen values, while the magnitude of error is inversely proportional to it.

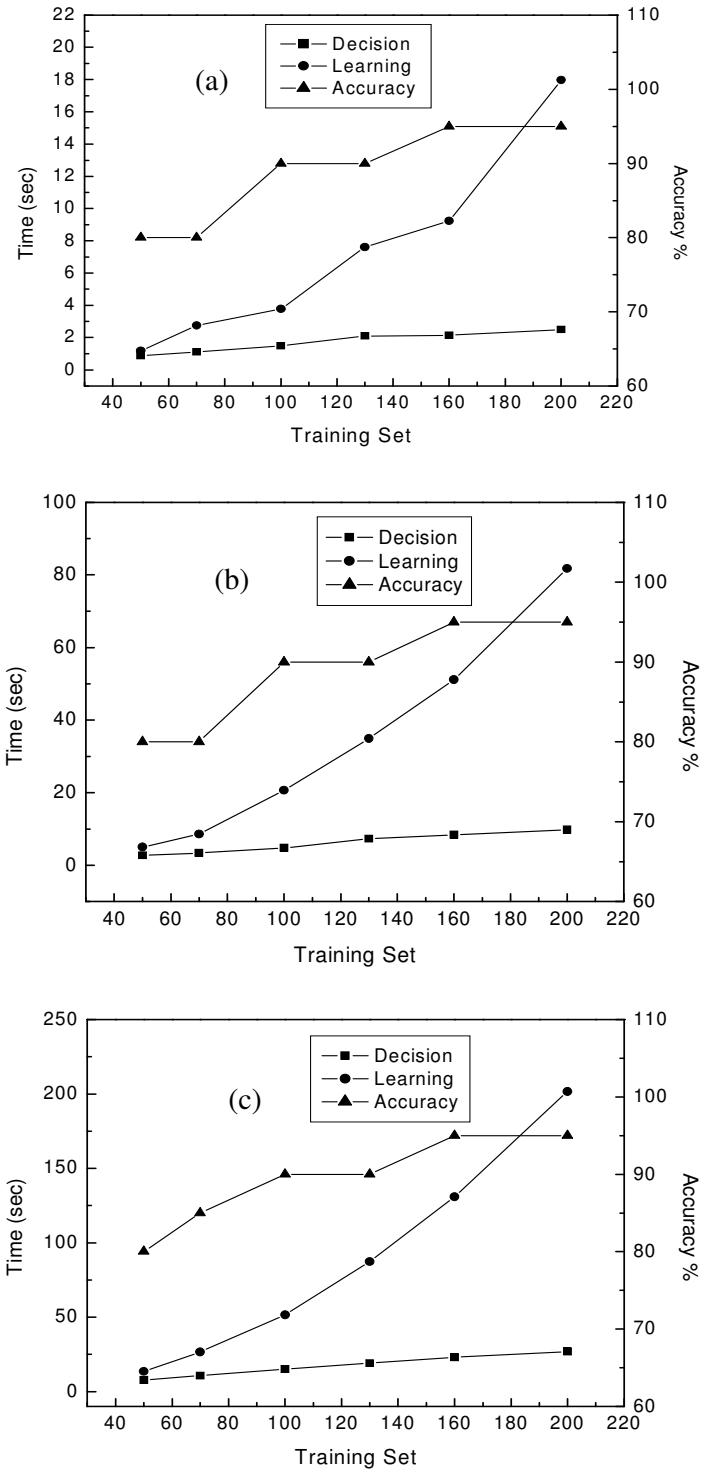


Figure-3.2 The resolution of training data sets are 90 x 100, 180 x 200 and 270 x 300, of PCA in (a), (b) and (c) respectively.

Examination of Figure-3.2 clearly shows that maximum difference in decision and learning time is 174.68 seconds, 71.96 seconds and 15.46 seconds for resolution of 270 x 300, 180 x 200 and 90 x 100 respectively, the largest observed is for highest resolution having number of images in the TS. This difference decreases when the system uses less resolution with small TSs and it reaches to its minimum value when system employs 50 images in the TS having resolution of 90 x 100. Furthermore, Figure-3.2 (a) shows that the maximum accuracy of PCA based system is 95%. This accuracy can be achieved by proposed system with various combinations of leading Eigen values as shown in Figure-3.3, although decision and learning time is small for 50 Eigen values. Similarly for TS having images 180 x 200 resolution as shown in Figure-3.4 and for 70 Eigen values in TS having images 90 x 100 resolution as shown in Figure-3.5.

On the other hand, the decision and the learning time shown in Figure-3.4 reflect almost a linear relationship when compared to the decision and the learning time of image 270 x 300 and 90 x 100 resolutions as depicted in Figure- (3.3) & (3.5). The plot of the Figure-3.2 clearly demonstrates that all TSs touches the same accuracy as that of 180 x 200 resolutions.

A comprehensive view of best possible combination of decision and learning time against accuracy of PCA and proposed system is shown in Table-3.3. The data of the table shows that for PCA, the accuracy remains the same for different TSs, whereas for proposed system number of LEVs plays a crucial role. Entries in Table-3.3 show best possible combination of the system variable.

Table-3.2 Training data sets showing samples per subject and LEVs for training.

Data Set Name	Total Image	Sample per Subjects	LEVs for Training
S_1	200	20	10
S2	160	16	9
S3	130	13	8
S4	100	10	7
S5	70	7	6
S6	50	5	5

Table-33 Decision and learning time varies with error rate associated with different image resolutions.

IR	Accu. %	PCA_DT in sec	PCA_LT in sec	min. TS	AFP_DT in sec	AFP_LT in sec	min. TS	LEVs
270 x 300	100	NA	NA	NA	14.938	52.984	100	100
270 x 300	95	22.96	130.92	160	3.375	4.9531	50	20
270 x 300	90	14.937	51.438	100	3.45	9.9531	100	20
270 x 300	85	10.61	26.531	70	1.875	5.8281	130	10
270 x 300	80	7.672	13.469	50	2.4066	9.3594	200	10
180 x 200	95	19.094	51.125	160	2.5625	4.3281	50	50
180 x 200	90	16.432	20.641	100	0.5	0.8125	100	10
180 x 200	80	10.375	5.0313	50	0.4375	0.17188	50	3
90 x 100	95	2.14	9.2344	160	0.6719	1.0469	70	40
90 x 100	90	1.484	3.7813	100	0.234	0.48438	100	10
90 x 100	80	1.1094	2.75	70	0.156	0.078125	50	3

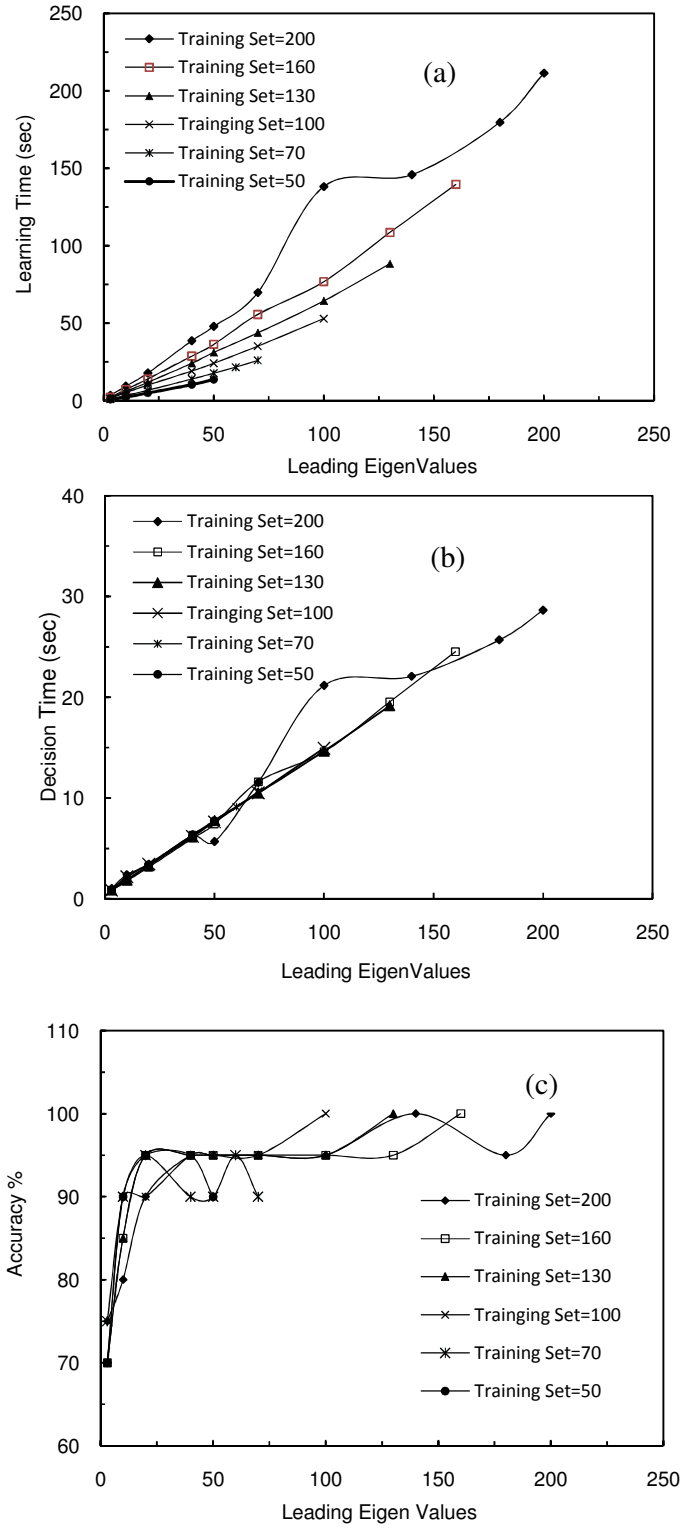
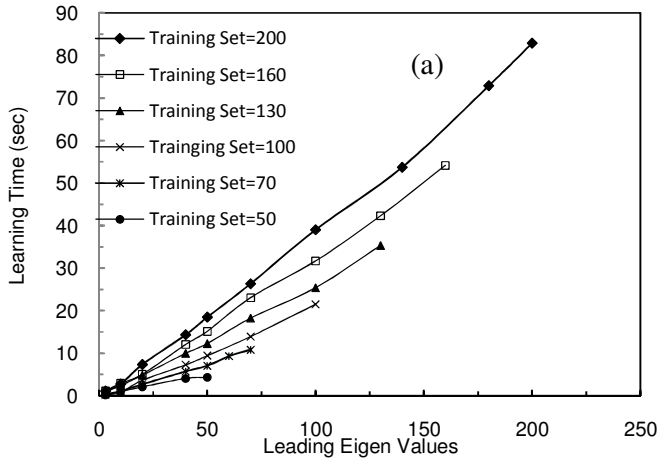
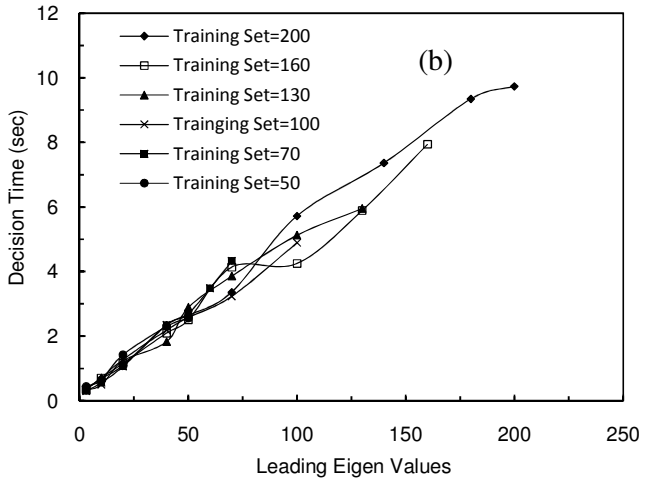


Figure-3.3 For the resolution 270 x 300, of proposed system in (a),learning time (b) decision time (c) accuracy of the system against leading Eigen values respectively.





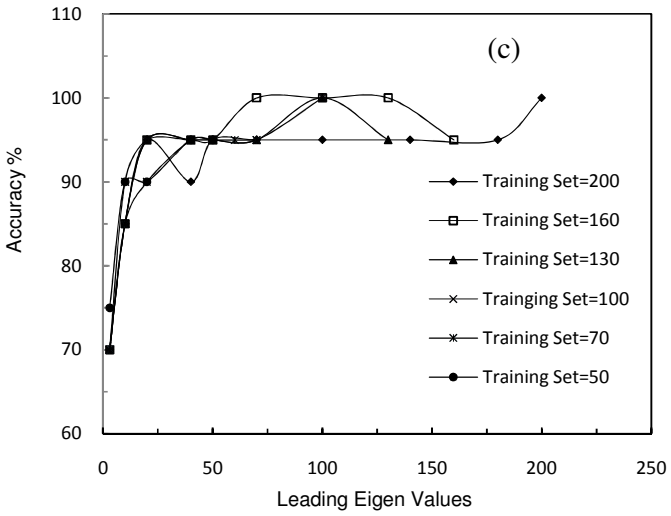


Figure-3.4 For the resolution 180 x 200, of proposed system in (a),learning time (b) decision time (c) accuracy of the system against leading Eigen values respectively.

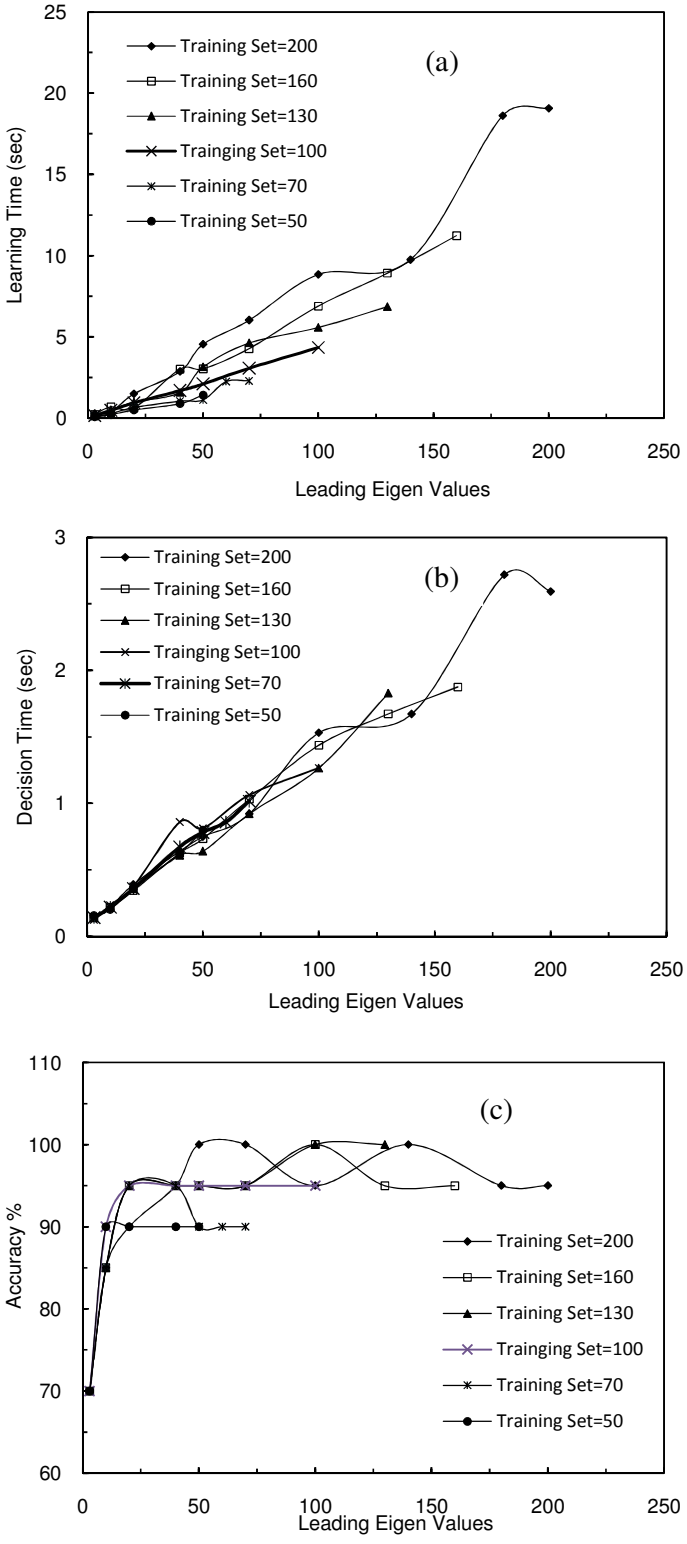


Figure-3.5 For the resolution 90 x 100, of proposed system in (a), learning time (b) decision time (c) accuracy of the system against leading Eigen values respectively.

Decision and learning time increases by increasing the accuracy for both the systems whereas the rate of increase in decision and learning time of proposed system is sufficiently less than a conventional PCA system. The maximum difference of decision time in PCA and proposed system is about 16 seconds with 95% accuracy for 100 images.

PCA provides 95% accuracy for the three said resolutions involved. Whereas, in this discussion it requires a minimum 2.14 second for decision and 9.23 second for learning when it is trained by using 90 x 100 resolution. On the other hand, proposed system requires 1.04 second for decision and 0.6719 for learning, when proposed system is trained with 70 images and uses at 40 LEVs.

Thus, the optimum selection of principal components and TS, proposed system gives efficient face recognition with PCA. In general, it has been demonstrated that proposed system provides time efficient decision for various image resolutions. Further, the technique provides a definite decision for a real-time face recognition system even for high resolution images.

Over thirty combinations of training and testing data sets were designed for testing of accuracy of the decomposition-based method and the modified Fast PCA as shown in Figure-3.6. Selection of images in training and testing data sets were random to assess the effectiveness of the proposed technique.

The test data sets had two categories of images. First category was associated with subjects who were present in the training data set but with different facial characteristics. The second category of images had subjects who were not present in the training data.

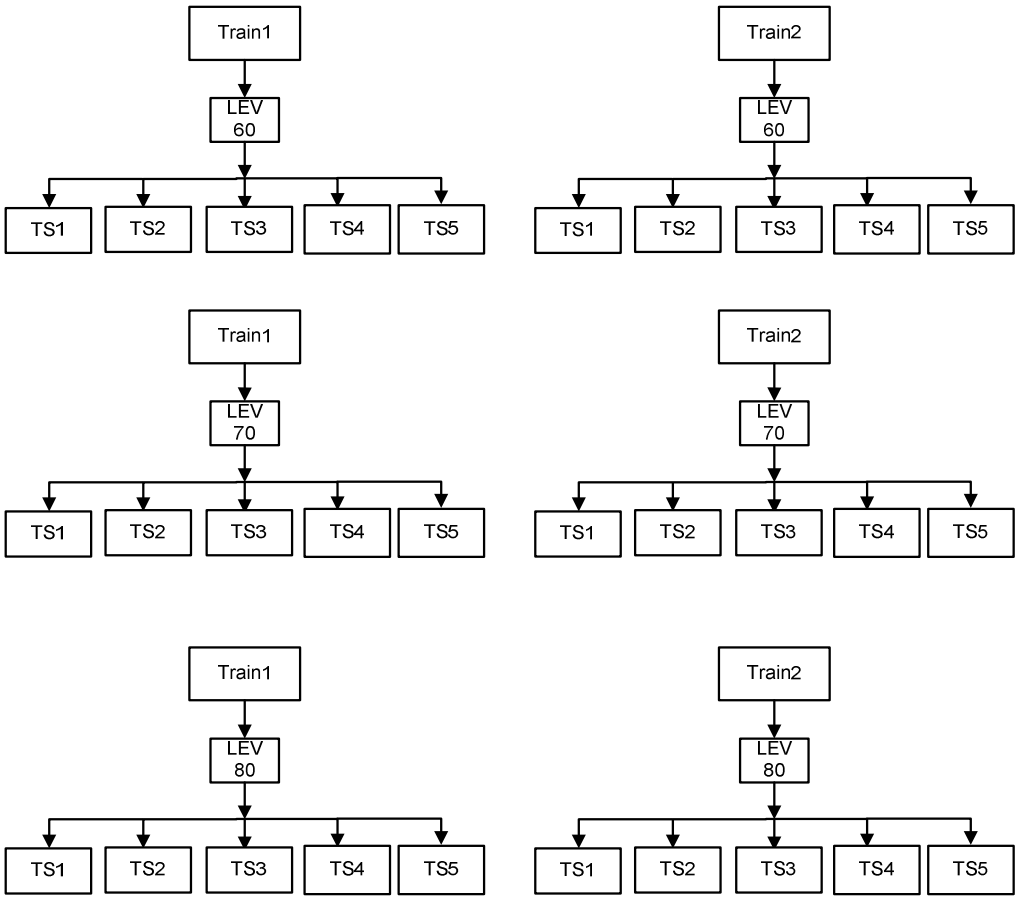


Figure-3.6 Different combination of training and testing data sets.

Assuming that TS1 represents data set having 100 images, TS2 for 200 images and so on. Since each data set was constructed randomly from the image library thus, the probability of two data sets to be 100% identical is rare. On the other hand, Train1 denotes training data set having 450 images and Train2 represents 225 images

Based on the chosen testing and training data sets, six sub trees were constructed for different LEVs as illustrated in Figure-3.6. Experiments were then performed for each combination by employing both decomposition and the proposed techniques.

A face which is not related to the training subjects and is recognized at verification is treated as false acceptance and its value divided by unknown is treated as FAR. Similarly, a face which is rejected while its subject is used for training is known as false rejection and its value divided by known tests is treated as FRR. In face recognition the same value of FAR and FRR is called EER which depends on the threshold value of the classifier. EER usually depicts MER of a system. But it is quite possible, in some cases, that the value of MER may be lower than EER.

ROC curves for 225 and 450 training images are generated by applying Equation (3.9) as shown in Figure- (3.7) & (3.8) respectively. Both the figures clearly demonstrate that the proposed technique has relatively better MER than the decomposition method. The results of the experiments have been summarized in Table-3.4 and Table-3.5. Comparing the magnitude of MER and EER, it is evident that the proposed technique is significantly better than the decomposition method.

By using Equation (3.10), the sum of FAR and FRR for various threshold values have been evaluated and shown in Figure-3.9. The plot also represents least square fitting of experimental data.

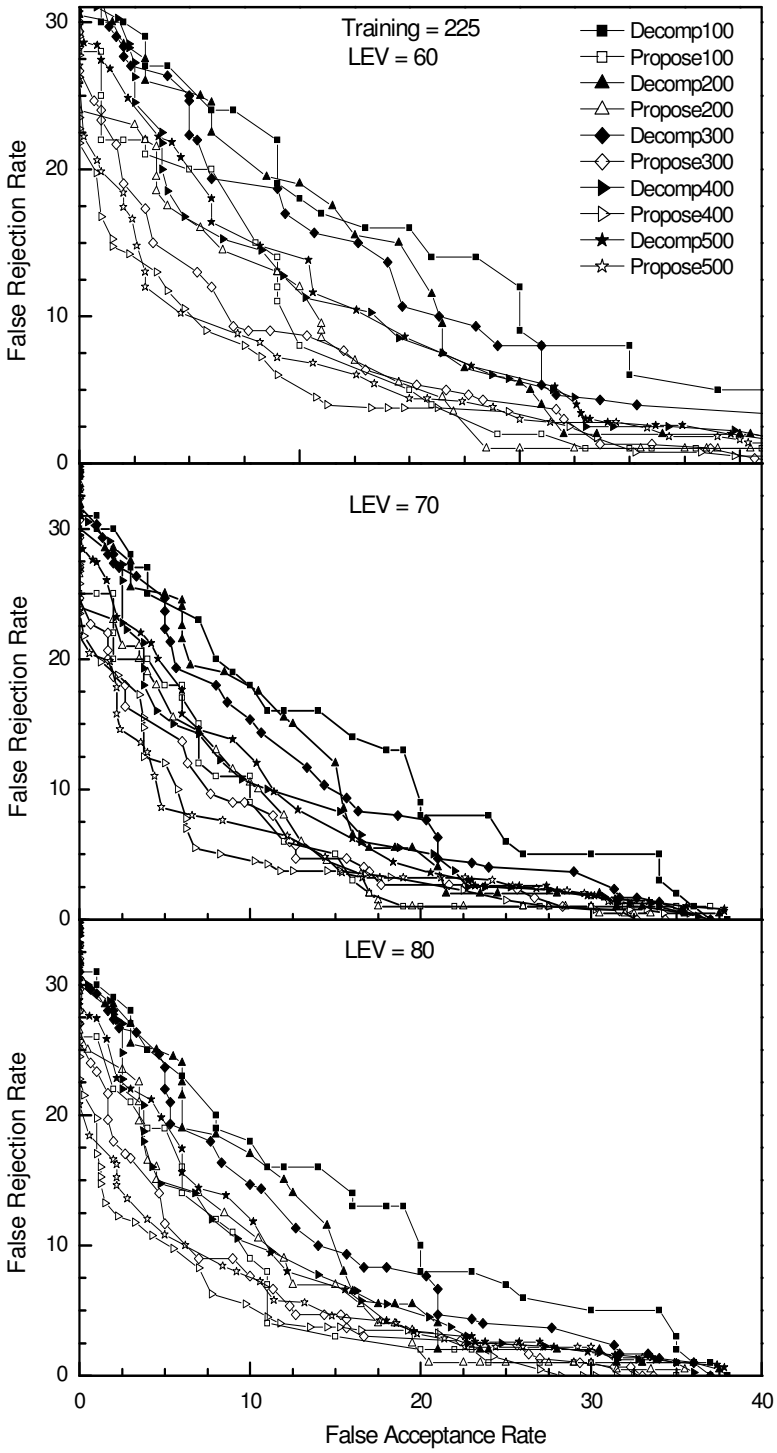


Figure-3.7 Comparison in false acceptance rate (FAR) vs false rejection rate (FRR) when system is trained with 225 images. Filled symbols represent decomposition method whereas empty symbols show the proposed technique.

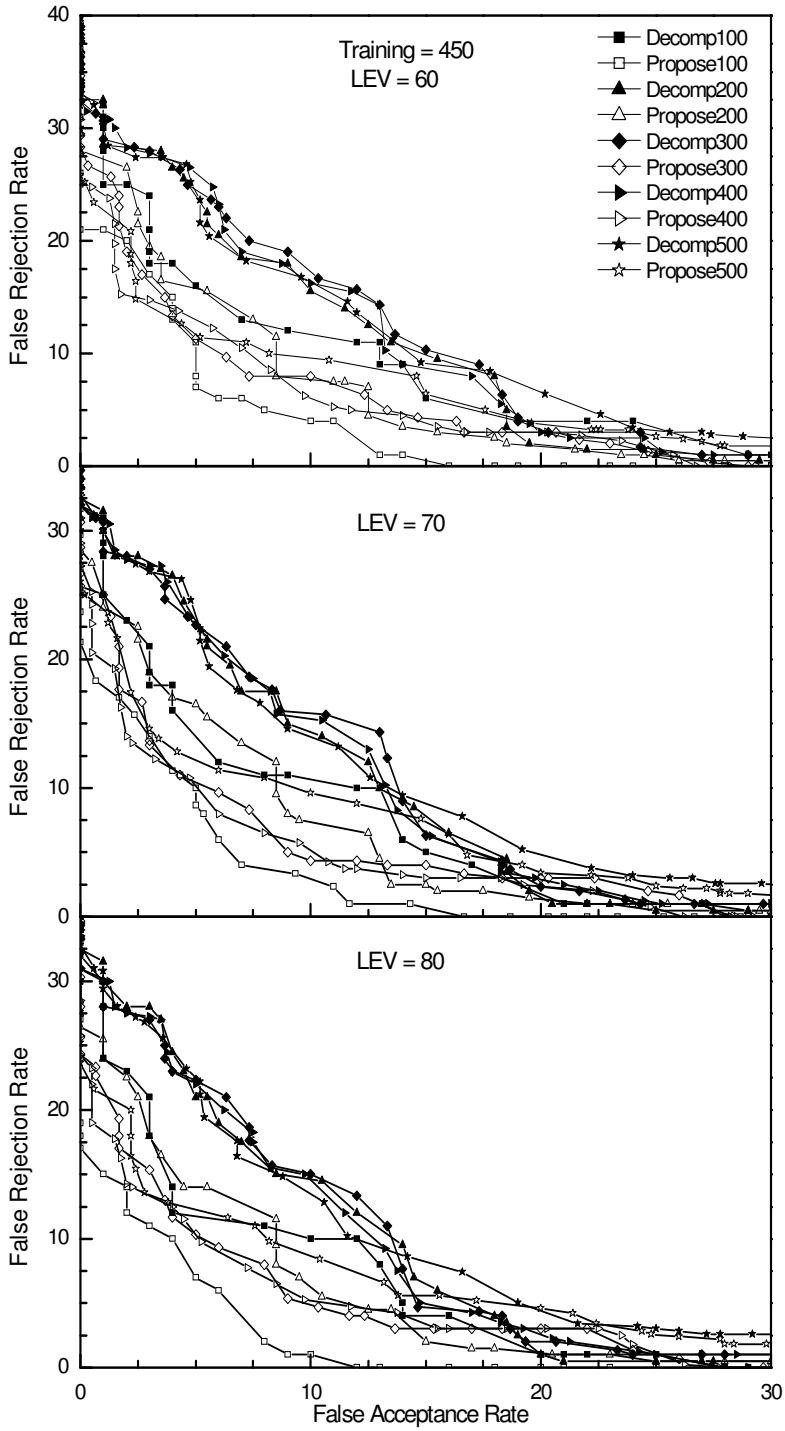


Figure-3.8 Comparison in false acceptance rate (FAR) vs false rejection rate (FRR) when system is trained with 450 images. Filled symbols represent decomposition method whereas empty symbols show the proposed technique.

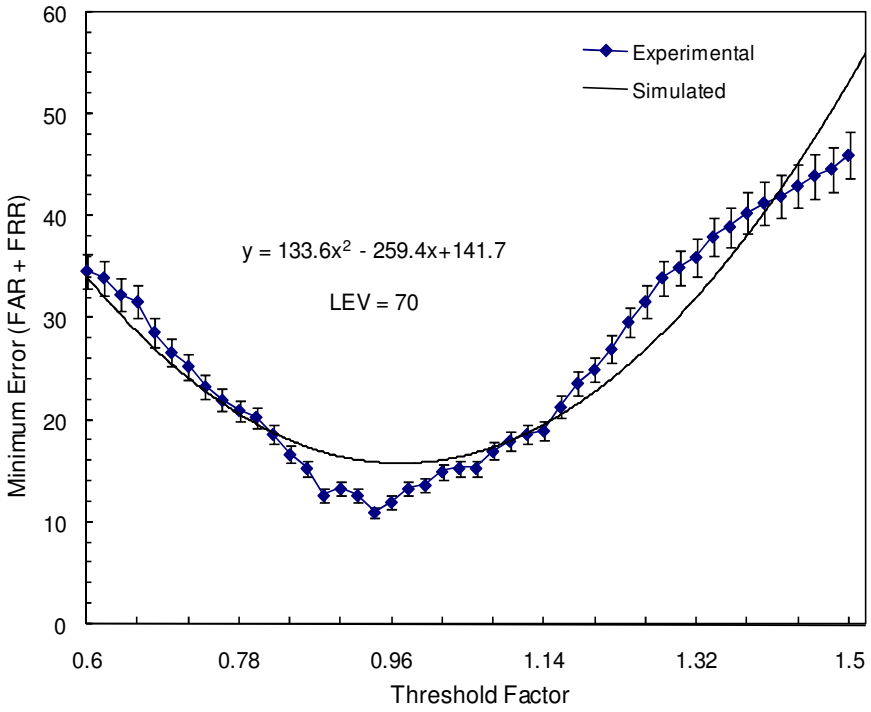


Figure-3.9 Minimum error as a function of threshold value of the proposed system. Solid line represents the curve fitting by least square method.

The simulated curve has a convex profile which defines MER for a given number of LEVs. Whereas, MER variation as a function of LEVs for different threshold values is shown in Figure-3.10. The plot of Figure-3.10 (a) represents MER of a system designed by using the decomposition method with threshold value as a variable whereas, Figure-3.10 (b) represents the same for the proposed technique. The decision time as a function of LEVs proposed system is presented in Figure-3.11.

Since the training sets involve 225 and 450 images, consequently the maximum Eigen vectors of the reduced covariance matrices are 225 and 450 for Train1 and Train2 respectively (Turk 1991). In the proposed system, LEVs are computed in descending order because in the proposed algorithm the subtracting term of step (vi) is minimum for the first leading Eigen vector and it increases for the upcoming Eigen vectors. It is observed that the first LEV is of the order of 11, second of the order of 10 and from three to eleven they are of the order of 9. Since for one combination of LEVs, there is an associated ROC curve, a range of ROC curves have been generated starting from 1.4% to 40% of the total Eigen values. It is observed that below 1.4% MER is unacceptably high and by increasing LEVs above 40% does not give a tangible improvement in the system performance.

On the basis of generated ROC curves, MER, the value of *p*, decision time and number of LEVs are gathered, called offline mode of operation. It is observed that the magnitudes of FAR and FRR are reciprocal in nature, as shown in Figure- (3.6) & (3.7). Ideally, a point should exist on each ROC curve where both the errors (FAR & FRR) are the same and minimum. But most of the time, it is not the case. Therefore, MER is proposed to characterize a system.

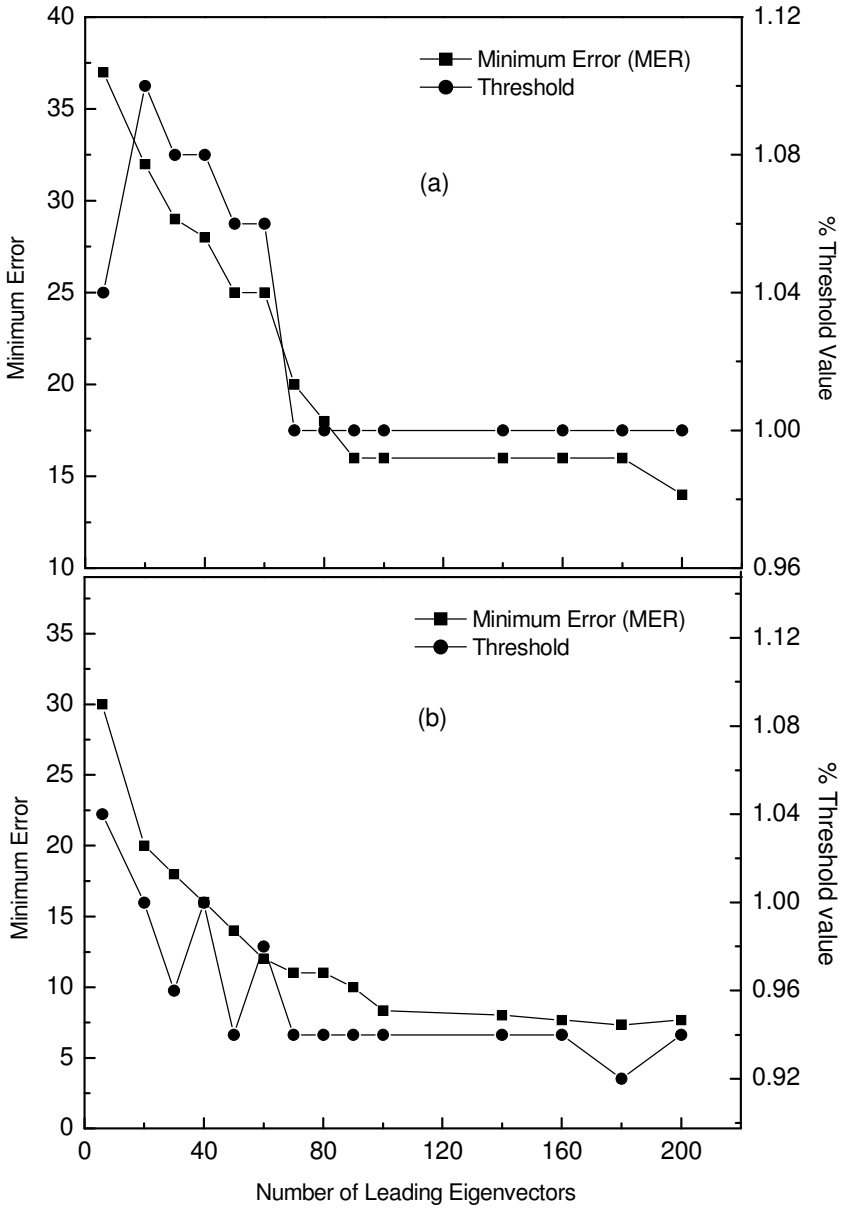


Figure 3.10. Minimum error as a function of leading eigenvectors (a) for decomposition and (b) for the proposed system.

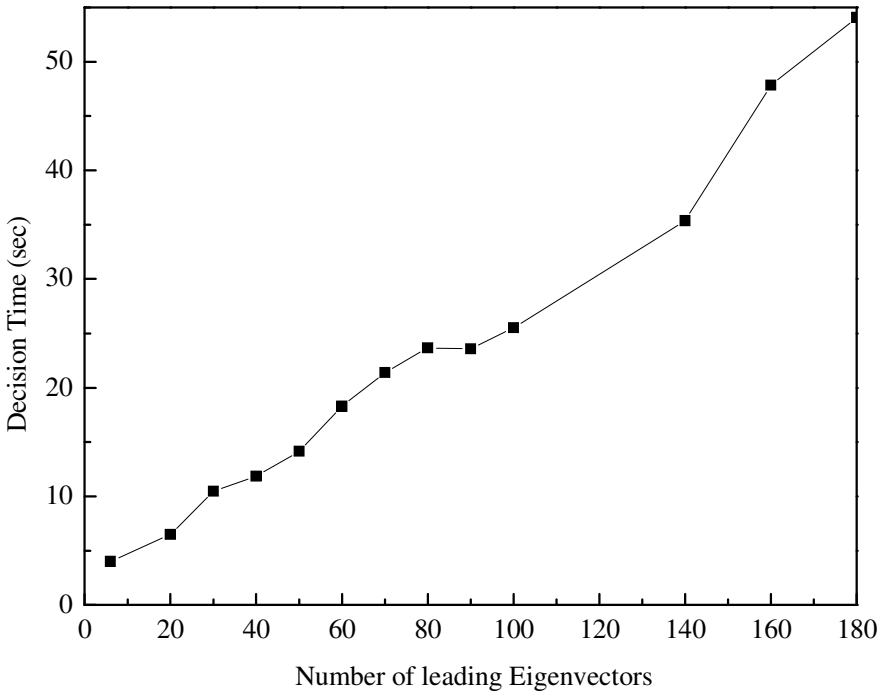


Figure-3.11 Variation of decision time against leading eigenvectors when 450 images are used in training and 100 images in testing.

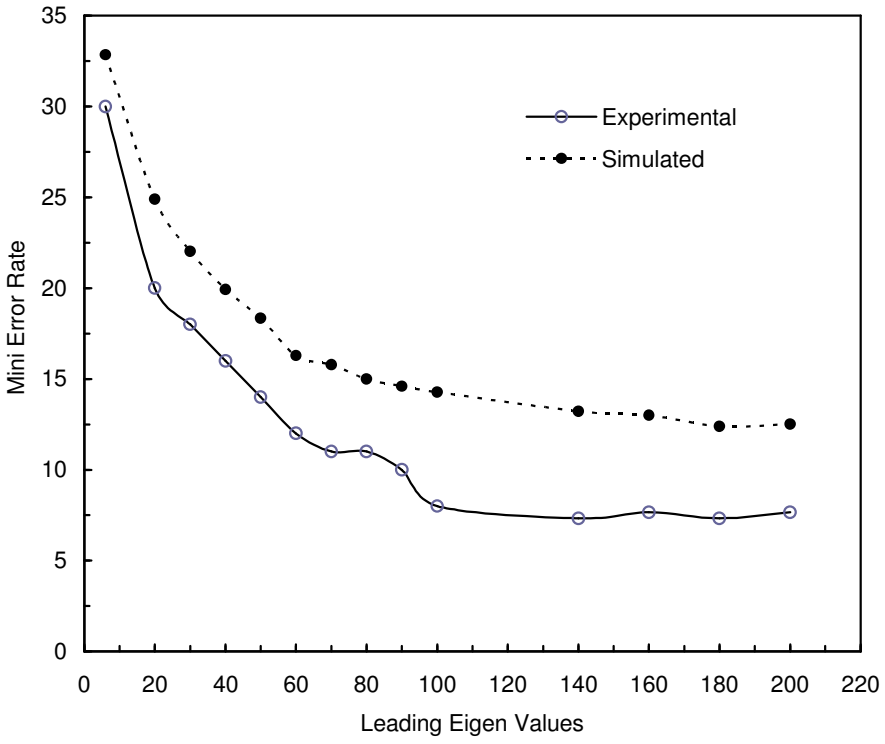


Figure- 3.12 Simulated and experimental minimum error rate vs number of leading Eigen vectors.

If there are more than one MERs, preference will be given to that MER which has lower FAR, to minimize the security risk of a recognition system.

The magnitude of MER has got an inverse relationship with number of LEVs as evident from Figure-3.10. However, it is observed that MER saturates between 20% to 30% of LEVs. On the other hand, increasing the number of LEVs increases the decision time of a system.

Thus, it is necessary to choose an optimum number of LEVs without compromising on the accuracy as well as on the decision time of a system. Further, it is observed that the value of *p* fluctuates for lower number of LEVs, but it stabilizes when the system includes approximately18% of LEVs. This once again suggests that an appropriate number of LEVs is required to define a stable system.

From Figure-3.9 it is evident that MER of the proposed system occurs at $p = 0.95$. Examination of the figure revealed that both the global and the simulated minimum appear at the same value of *p.* But, the simulated minimum is slightly higher than the experimental one. Figure-3.12 shows a comparison between experimental and simulated MER as a function of LEVs. An important point to note is that both the curves show a similar profile, but with a finite difference. This demonstrates the validity of simulated data but, of course, with a finite error. To remove the error between the experimental and simulated characteristics following correction factor is proposed: $ceil(mean(\sum_{i=1}^{n}|T_i - E_i|))$ where n represents total data points, T theoretical and E experimental value at i^{th} point respectively. The proposed relationship works well for various training sets as shown in Figure-3.13. Furthermore, the observed root

mean square error, as calculated by (10), between the simulated and the experimental data for various LEVs (6-200) was less than 1.1.

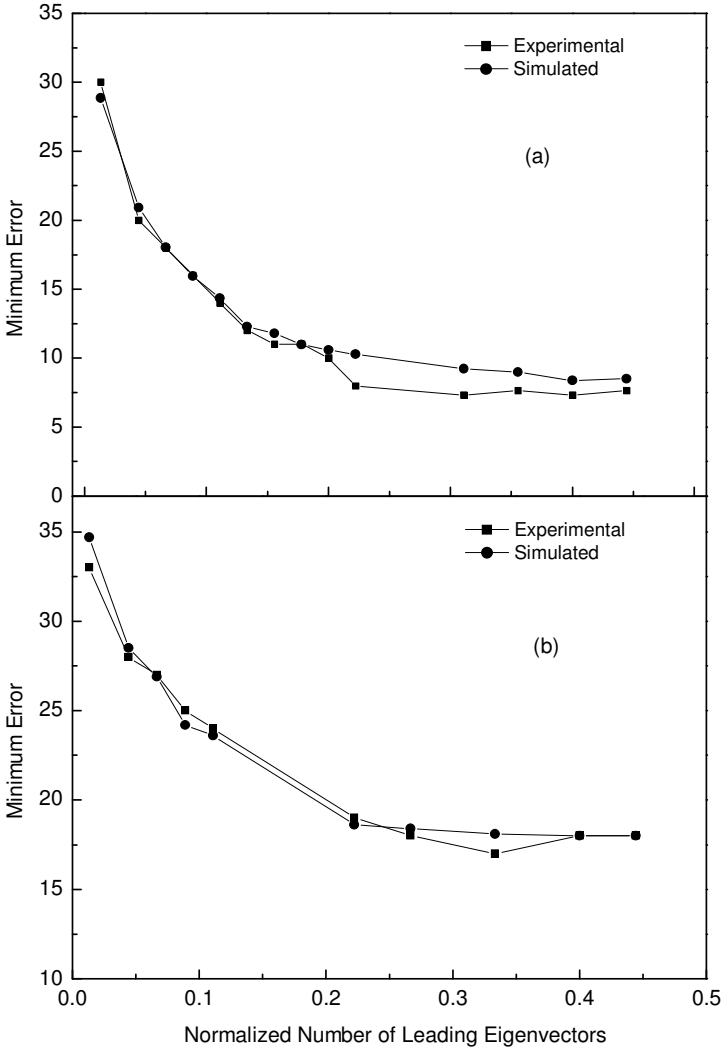


Figure-3.13 Experimental and simulated data against normalized number of leading Eigen values (a) for training set of 450 images (b) for training set of 225 images.

Table-3.4 Comparison between decomposation and the proposed technique when system is trained with 225 images.

LEV 70	Decomposition		Proposed		% Improvement	
Test Set	MER	EER	MER	EER	MER	ERR
TS1	18	20	11	12	38.9	40.0
TS2	21	24	16	17	23.8	29.2
TS3	21	25	14	15	33.3	40.0
TS4	21	25	14	14	33.3	44.0
TS5	24	24	17	19	29.2	20.8

Table-3.5 Comparison between decomposation and the proposed technique when system is trained with 450 images.

.LEV 70	Decomposition		Proposed		% Improvement	
Test Set	MER	EER	MER	EER	MER	EER
TS1	27	30	19	21	29.6	30.0
TS2	24	27	18	21	25.0	22.2
TS3	25	25	17	18	32.0	28.0
TS4	21	21	12	13	42.9	38.1
TS5	21	21	13	16	38.1	23.8

This value is well within acceptable error margin for a face recognition system.

$$\sqrt{\frac{\sum_{i=1}^{n}(E_i - T_{adj,i})^2}{n}} \qquad \ldots\ldots \text{Eq (3.12)}$$

In an online mode of operation, let *F* matrix be populated by the information based on Figure-3.6 & Figure-3.7 and assuming a maximum allowable decision time of an application is 0.3 seconds per frame. Then the system adaptively chooses $h = 100$, such that it provides decision in 0.25 second per frame with an accuracy of 92%, whereas a system defined by the same variable but using decomposition technique provides decision in 0.83 second/frame with an accuracy of 84%.

3.8 Summary

This chapter presents a new architecture to implement an optimized decision support system based on reduced dimensionality of covariance matrix in Eigen space. The leading components of reduced matrix are computed by using modified Gram-Schmidt orthogonalization (MGSO). Use of MGSO in fast PCA has improved the convergence and accuracy of a face recognition system especially for high dimensional images. To optimize the performance of a face recognition system, receiver operating characteristics are simulated with least square fitting. Convex optimization is applied for optimum selection of system variables. It has been shown, by comparing the simulated and experimental data, that the root mean square error of the proposed system is less than 1.1. It is shown that the system is robust enough and it works reasonably well for images acquired under varying lighting conditions.

CHAPTER 4

HOUSEHOLDER IMPLEMENTATION ON FPGA

4.1 Introduction

For hardware evaluation of eigenvalues, CORDIC-based Jacobi algorithm (CJA) is one of the best reported implementations on FPGA (Bravo 2006). Basically, CORDIC implements trigonometric functions in hardware which are commonly used to evaluate Eigen values (Bravo 2006, Ray 1998). However, CORDIC employs approximation in vector rotation to estimate trigonometric functions. On the other hand, Householder (HH) method uses square root instead of trigonometric function for the evaluation of Eigen values which made it more accurate than Jacobi algorithm (Press 1992, Golub 1996, Sajid 2010-b).

Floating-point implementation of a square root is computationally intensive than its integer based implementation (Soderquist 1996, Soderquist 1997). Many modern microprocessors have high atency to throughput ratio (Kwon 2009). Floating-point square root accuracy is as good as that of an analytical solution (Wilkinson 1962, Ortega 1963, Burden 1997). However, floating point implementation requires more space on chip and consumes higher number of cycles for the same operation (Sajid 2008-a, Oberstar 2007, Liao 2000).

HH is considered an efficient method for eigen solutions, because it terminates definitely after n-2 iterations, where n is the size of an input symmetric matrix (Press 1992). It is worth mentioning that the performance of HH for the evaluation of Eigen

values is biased towards the truncation of least significant digits during a looping process (Wilkinson 1962, Wilkinson 1965).

It is an established fact that software based applications are slow but save system resources, whereas hardware based solution provides time efficient systems but may require resources beyond the maximum feasible limits. Thus, a co-design methodology could provide a cost and time effective solution for HH implementation on FPGA. Two FPGA based architectures have been proposed for efficient HH implementation. Performance of these architectures have been tested and analyzed by considering time, power and precision parameters.

4.2 Householder on FPGA

HH algorithm can be made time-efficient by employing an intelligent hardware/software co-design model (Pellerin 2003, Lee 2009). In this model, HH is divided into hardware and software processes. First in first out (FIFOs) buffers are used to facilitate the flow of data between different processes. Deadlock, which is a common problem in the implementation of a FIFOs-based steaming model, is avoided by adjusting the FIFOs length and communication modes. Communication modes could be blocking or non-blocking. In blocking mode, FIFO is compelled to produce output data in the same sequence as it was supplied. On the other hand, a non-blocking mode does not ensure the sequence of data and may produce inaccurate results. Thus, blocking mode of FIFOs are used to implement HH.

Two different architectures for the implementation of HH have been developed using VisualC, embedded development kit (EDK) and integrated software environment

(ISE). The performance of these architectures is then evaluated by using synthesis reports, data sheets and error analysis techniques (Shang 2002, Klein 2005).

4.2.1 Floating Point Architecture

Figure-4.1 shows a floating point architecture for the implementation of HH on FPGA. It is comprised of an application layer, a real-time operating system (RTOS) layer and a hardware layer. Image data producer (IDP), data consumer (DC) and Eigen values generator (EVG) are the software processes as evident from the architecture.

IDP reads images stored in FPGA auxiliary memory and computes a co-variance matrix which is used by the EVG as an input data. The EVG customizes the HH according to the processor architecture, so that it can be translated into low level instructions by the RTOS.

In this architecture, integer operations are computed by the processor itself, while the floating operations are directed to the floating point unit (FPU) as illustrated in the figure. This technique is adopted because ALU does not support floating operations. As per Xilinx technology, fabric core bus (FCB) is used to interface FPU with auxiliary processing unit (APU, APU 2007).

It is obvious from Figure-4.1 that a modular approach has been adopted to implement HH on FPGA. It allows modification in HH at application level without requiring any change in the hardware. Further, the architecture ensures evaluation of Eigen values as per IEEE-754 standard of single precision (Kahan 1997).

4.2.2 Fixed Point Architecture

Contrary to the approach discussed in Section 4.2.1, an architecture could be designed by implementing a significant portion of HH on hardware. In such an architecture, Eigen values can be generated by using hardware resources of FPGA. This process, by its very nature, will be time-efficient compared to the architecture shown in Figure-4.1. However, the design requires more resources and power, and may not be feasible for a portable system. Thus, a co-design methodology has been proposed. In this co-designed architecture computationally intensive part of HH is divided into two parts: a) software namely FxHH and b) hardware namely FxSqr as shown in Figure-4.2.

FxSqr in Figure-4.2 was first implemented by using ARM algorithm given in (ARM 2009) for fixed point square root realization. It was noted ARM algorithm provides accurate answer for those integers whose square root is a whole number but for answer containing fraction its accuracy deteriorates. To overcome this bottleneck, the FxSqr was modified with non-restoring division algorithm (Piromsopa 2001) which provided an accuracy up to ten decimal places.

Fixed-point hardware with appropriate software support is often used for the low-cost implementation of algorithms requiring floating-point operations. This alternative usually provides accuracy very close to that of floating-point hardware. Furthermore, the fixed-point accuracy heavily depends upon the operand values and how properly the *q.n* format for fixed-point operations is adjusted. Small or extremely large values for the operands may produce minor errors for complex operations due to the rounding up and the truncation of least significant digits (Wilkinson 1962, Ortega 1963, Burden 1997). However, real-time portable applications emphasize time and

power efficient solutions since minor errors in the answer do not often affect the quality of the results significantly for relevant applications.

The Newton-Raphson, SRT-redundant and non-restoring division techniques form the bases of the three main categories of algorithms used for the evaluation of the square root. The Newton-Raphson method has self-correcting capability and provides relatively better precision but requires an initial guess using a seed generator while it converges quadratically (Karp 1997). Furthermore, it is based on an iterative approximation technique which requires frequent multiplications so a pipelined implementation of Newton's method requires large multipliers which may not be a feasible option. On the other hand, the SRT-redundant and non-redundant algorithms are both recursive in nature and their implementation is costly especially for higher order radices (Piromsopa 2001).

Pipelining is usually adopted to make a system time efficient but requires extra effort to resolve timing and data hazards. Pipelined implementation of the square root using non-restoring algorithm and fixed-point arithmetic faces data dependence hazard (Sajid 2010-b). Then, a novel pipelined architecture has been presented to avoid data dependence hazards by employing a modified non-restoring division algorithm. This architecture shows a time and power efficient evaluation of the square root as compared to floating-point based architectures.

4.2.2.1 Pipelining of Square Root

In digital applications, pipelining is the most common parallel scheme used to improve the throughput. A complex task is divided into subtasks which are implemented independently in their respective execution stages.

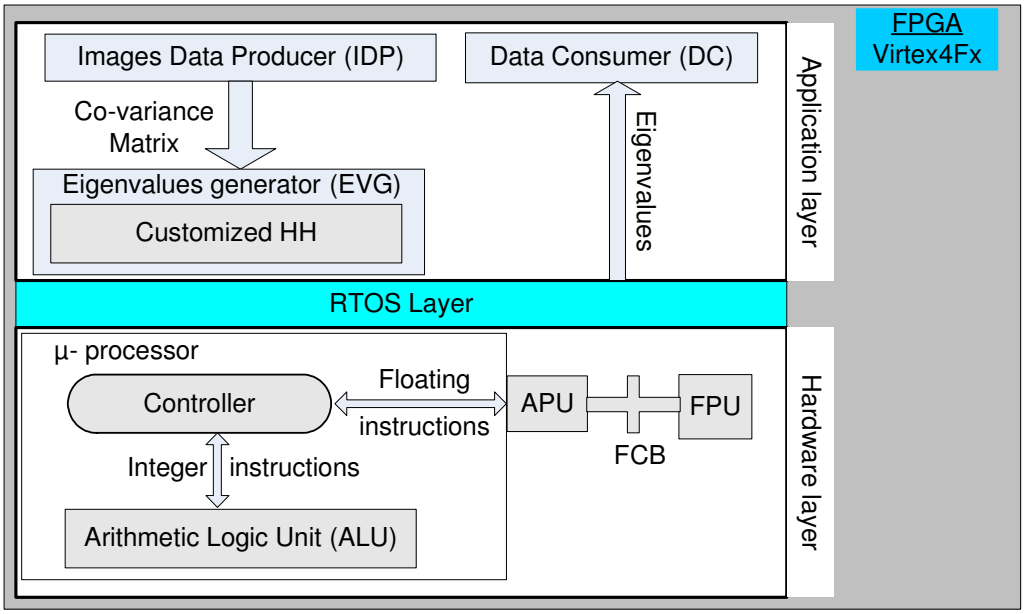


Figure 4.1 HHM implementation on FPGA using floating point architecture.

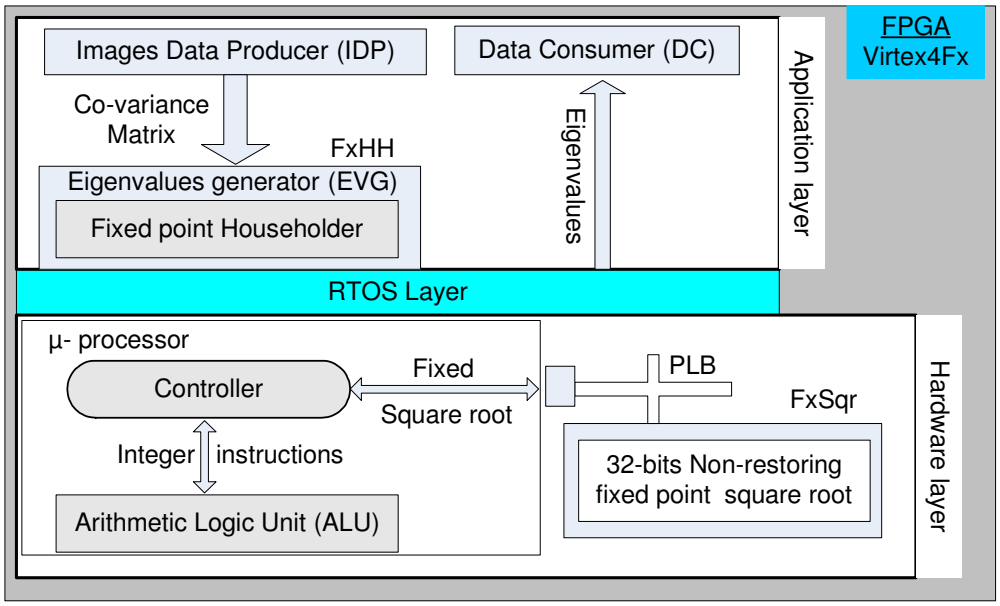


Figure 4.2 Fixed point implementation of HHM using co-design architecture.

It is required that the output of each subtask become available to the subsequent unit as and when needed. A write-after-read (*WAR*) hazard may arise if a stage in a pipelined system tries to read data from a register written by an instruction that follows in program order.

The non-restoring division technique can be used to evaluate the square root without restoring the remainder of a division (Mailloux 2007). It generates one bit of the result iteratively using two bits of data. Let $D = D_N D_{N-1}.....D_1$ represent an N-bit radicand of an unsigned number. By using the non-restoring algorithm, the square root of D may be represented by $Q = Q_{N/2}\ Q_{N/2-1}....Q_1$ after the $N/2^{th}$-iteration. The remainder $R = R_{(N/2)+2}\ R_{(N/2)+1...}\ R_1$ has $(N/2)+2$ bits. The relation among R, D and Q is $R = D-Q^2$, where D is less than $(Q+1)^2$ because $R < 2Q+1$. A functional block representation of the non-restoring based square root evaluation is given in Figure-4.3. The figure shows that there are only addition/subtraction and shift operations involved in the evaluation of the square root.

The conventional non-restoring algorithm was implemented for XC2VP2 Xilinx FPGA and the results were simulated in Integrated Simulated Environment (ISE) 8.2 and ModelSim 5.7 XE. Figure-4.4 shows the post layout simulation for a 2-stage pipelined implementation. Instability in the q_2 signal representing Q_2 was observed, which is associated with the fact that a register is receiving data from a stage while the next stage is trying to read the same register in the same clock cycle (this represents a *WAR* hazard).

a) Algorithm:

To overcome this *WAR* hazard, we modified the design as shown in Figure-4.5. *k* is a variable related to the bit-width of the *R* and *Q* registers, as required by the algorithm. With this modification, the '*if condition'* of the conventional version (Li 1997, Piromsopa 2001) has been replaced with a simple NOT gate. The shift operation, which is an integral part of the non-restoring algorithm, works well in the non-pipelined implementation where only one *Q* register is required as shown in Figure-4.3.

In the proposed pipelined implementation, an arrangement has been made for the proper placement of bits in the *Q* register from different wires in the same clock cycle without a shift operation. This provides stable and correct data for subsequent stages in the pipeline without losing clock cycle.

Figure-4.6 shows a conceptual flow of the pipelined implementation for the modified non-restoring algorithm. However, more *Q* registers for a pipelined implementation are required and their exact number depends upon the number of stages involved in the pipelined system. Two bits are fed to each stage from a *D* input register whose length is equal to *N* radicand bits. In stage-1, the two most significant bits (MSBs) are directly connected to M_1 and there are no pipeline registers. In stage-2, there are three pipeline registers, five pipeline registers in stage-3, and so on.

The circuitry associated with the M-modules of Figure-4.6 is shown in Figure-4.7. In this figure, S_n is a computing unit (ALU) and is labeled as S_1 for stage-1, S_2 for stage-2, and so on. S_1 gets the two LSB bits from the *D* register as its right operand, while the remaining bits of the left and right operands are initialized with one's and zero's

as shown in Figure-4.7; on the other hand, the LSB of the left operand of S_2 is fixed as one whereas the remaining bits are fed from the previous stage of the pipeline. S_1 is treated as a subtractor in stage-1 because its select bit is always low. S_2 could be a subtractor or adder depending upon the select bit whose value is defined by the MSB of the R_1 register.

b) Architecture:

To demonstrate the validity of the modified non-restoring realization, the concept in Figure-4.6 for *N=32* was implemented on XC2VP2 and XC3S100 Xilinx devices. The software tools are ISE 8.2 and ModelSim 5.7 XE. For N = 32, the number of stages defining the pipeline architecture is 16. In Figure-4.6, *Preg*1&2 act as padding registers while the r_2 receives data from stage-1. For n stages in the pipelined architecture, *2n-2 Preg* registers are required plus a register r_n which receives data from the previous stage, where n is the stage number. The r_n register is primarily used to store partial results from the Q_{n-1} register of the previous stage and provides the same data to the Q_{n+1} register of the subsequent stage in the pipeline. Therefore, the *WAR* hazard, which is related to read and write timing the Q register in Figure-4.4, has been solved.

To observe the operations in the proposed pipeline system, one stage at a time, consider Figure-4.7 with *N=4*. In the first clock cycle (T_1) the two MSB bits of the *D* register are transferred into the LSB positions of the right operand of S_1. The remaining k MSBs in the right operand of S_1 are fixed to zeros.

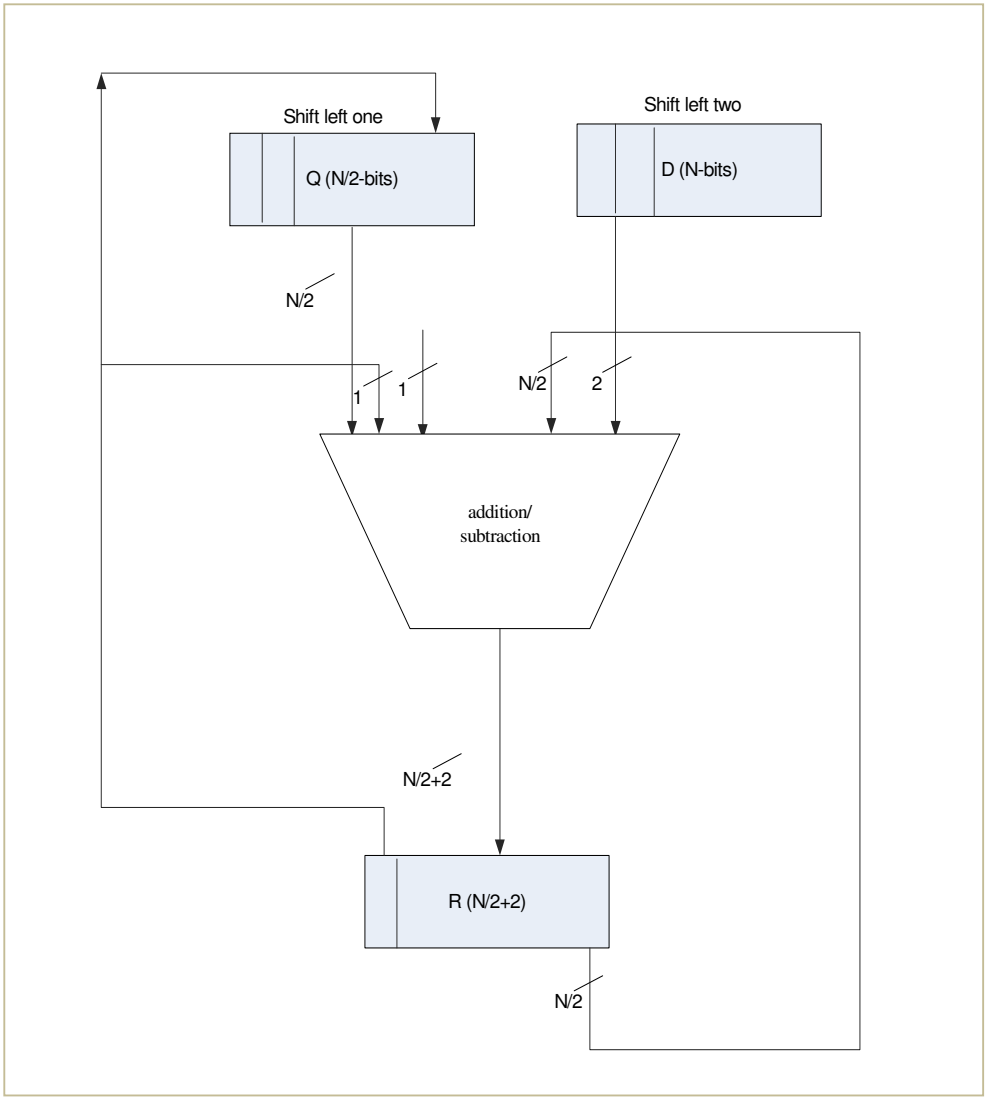


Figure. 4.3. Block diagram with non-restoring division.

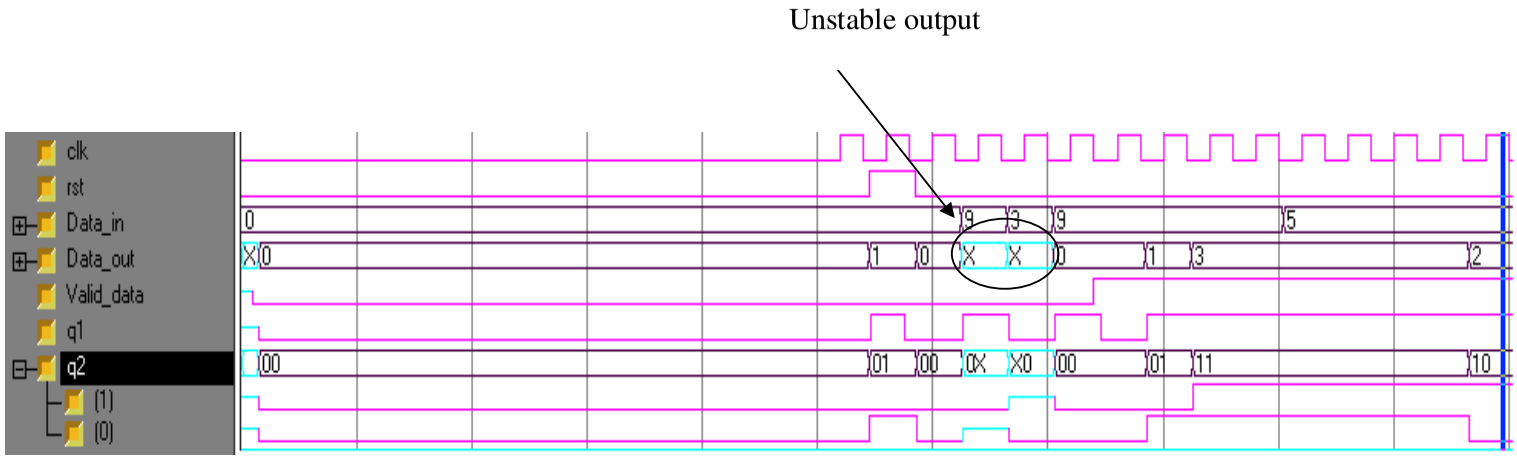


Figure.4.4 Unstable output due to a WAR hazard for the non-restoring algorithm (pipelined implementation with the XC2VP2 FPGA device).

```
Let D be an N-bit unsigned integer, Q be a k-bit unsigned integer and  R be a (k+2)-bit
unsigned integer
Declare R1=R2=.....Rk and Q1=Q2=…..Qk=0
// ( R and Q registers are initially filled with zeros to aviode the junk values).
#Define Con = Concatenation (x,y)
// (In HDL the bits of two variables x and y are concatenated using curly brackets by
comma i.e. {x,y})
loop i=1:k
  if (i=1)
    Ri=Con(k zeros, D[N-2(i-1)-1:N-2(i-1)-2]) - Con(k+1 zeros,1), sel=0
  else
     sel = Ri-1[k-1]
        if(sel=0)
            Ri=Con(R(i-1)[k-1:0],D[N-2(i-1)-1:N-2(i-1)-2])– Con(Q(i-1),R(i-1) [k-1],1);
               Qi=Con(Q(i-1)[i-1:1], Ri' [k+1]);
        else
            Ri=Con(R(i-1)[k-1:0],D[N-2(i-1)- 1:N-2(i-1)-2])+Con(Q(i-1) , R(i-1) [k-1],1);
               Qi=Con(Q(i-1)[i-1:1],  Ri' [k+1]);
       endif
  endif
endloop
```

Figure.4.5 Modified non-restoring algorithm for pipelined fixed-point based N-bit square root realization.

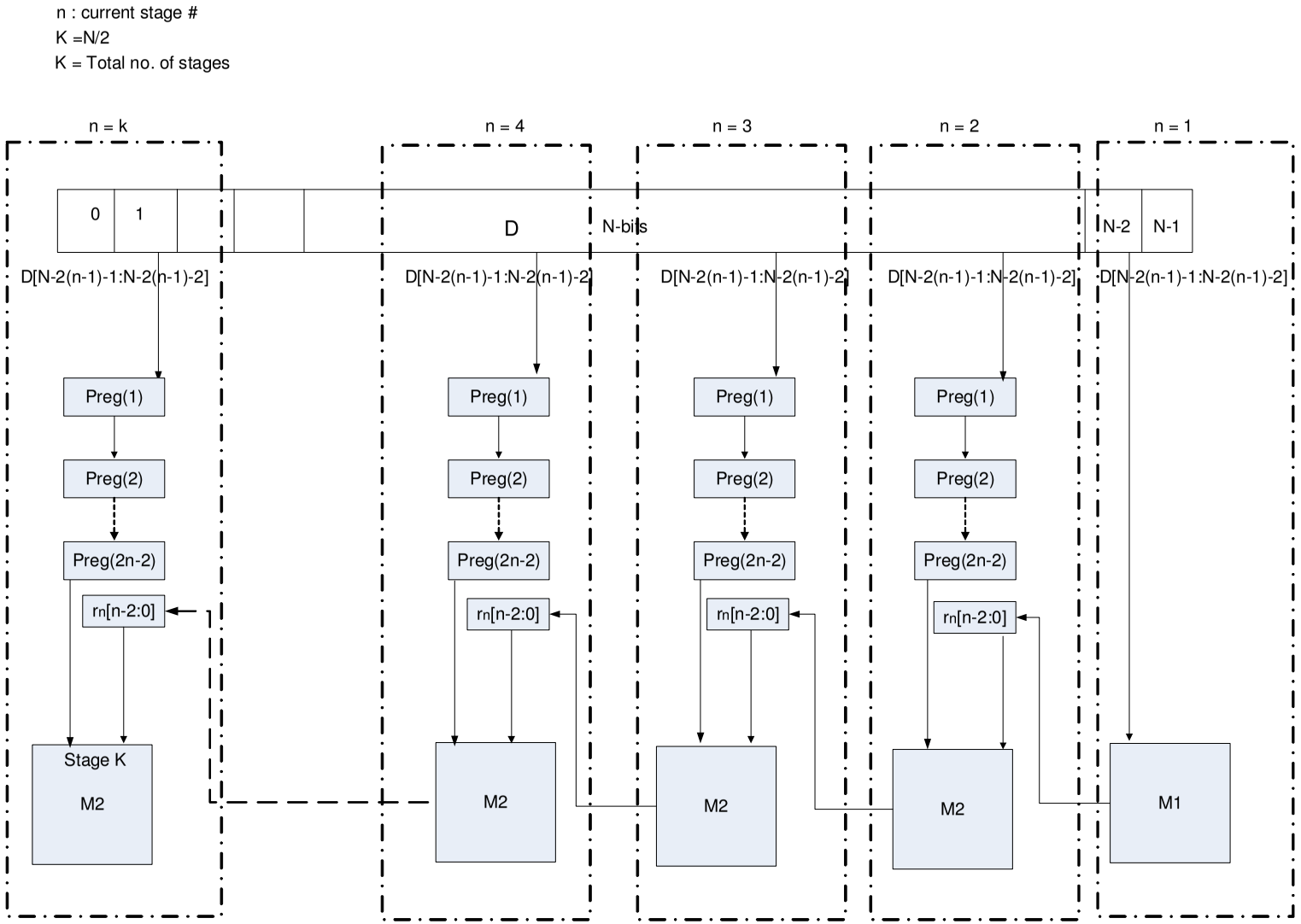


Figure.4.6 N-bit pipelined architecture for fixed point based square root implementation.

The k as a variable is equal to half of the radicand bits and $k+1$ bits are concatenated with 1 for the left operand of S_1. The result of the above operation is not immediately available from S_1 to R_1 because of the combinational delay in S_1. After that delay, $k+2$ bits from S_1 are available in R_1 but not placed in R_1. Furthermore, the MSB of S_1, after passing through an inverter, is available at Q_1 during T_1, but is not placed in Q_1 registers because the sequential units operate in a clock edge-triggered manner. Furthermore, in T_1 the two LSBs from the input D register are transferred into the $Preg$(1) which is a part of stage-2.

In T_2, $Preg$.(2) receives data from $Preg$.(1), Q_1 from the inverter and R_1 from S_1 which is available to S_2 and ready for computation because S_2 is a combinational unit. The LSB of the left operand for S_2 is always high and its next bit is defined by the MSB of R_1 (it determines addition or subtraction) after T_2 (because data from $Preg$.(2) is available to S_2).

In T_3, S_2 performs addition or subtraction and the pipeline register r_2 receives data from Q_1. Register r_2 contains just one bit and is part of stage-2. In this cycle, data is not placed in R_2 due to the delay in S_2, and similarly data from r_2 is not available to Q_2. In T_4, the final result will be computed and will become available in Q_2 after receiving data from S_2 and r_2.

Figure-4.8 shows a dry run of the proposed system. It explains the stage-wise changes in the R and Q registers per cycle, once the pipeline is filled. The answer is available at Q_4 for 8-bit data. Similarly extending the same logic, the answer for a 32-bit radicand will be available in Q_{16}.

Figure-4.9 (a&b) shows post layout results of non-pipelined and pipelined implementations of non-restoring algorithm on a Virtex2Pro (XC2VP2) device. Figure-7(c) represents once again pipelined implementation of the same algorithm on a Spartan3E (XC3S100) device. The post layout simulations shown in Figure-4.9 were generated after the place-and-route process of the Xilinx tools. In this figure, data_*in* and *data_out* show the input and output data, respectively. To demonstrate the validity of the proposed system, hundreds of randomly selected radicand values for the evaluation of the square root were used in simulations. The same output from various Xilinx devices shows that the proposed architecture is device independent. The total latency of the system was evaluated for various devices. This latency of the modified non-restoring algorithm is about 118 ns whereas it is 135 ns on the Virtex2Pro for the classical version, as shown in Figure-4.9 (a). Therefore, the proposed system demonstrates a 13% improved latency with respect to the classical system.

Table-4.1 shows time to evaluate a square root with non-pipelined and pipelined systems using Virtex2Pro from Xilinx and Flex from Altera. It is useful to consider power consumption along with the execution time because they are reciprocally dependent upon each other.

Various performance metrics have been reported for the evaluation of FPGA-based square root systems, but they are not comparable (Thakkar 2006-a, Thakkar 2006-b). Throughput (the inverse of the clock period for the slowest pipelined stage) is used as a performance metric. The throughput of a pipelined system is dependent upon number stages. Thus, a system comprising on larger number of stages will require higher resources and as a result it will consume more power.

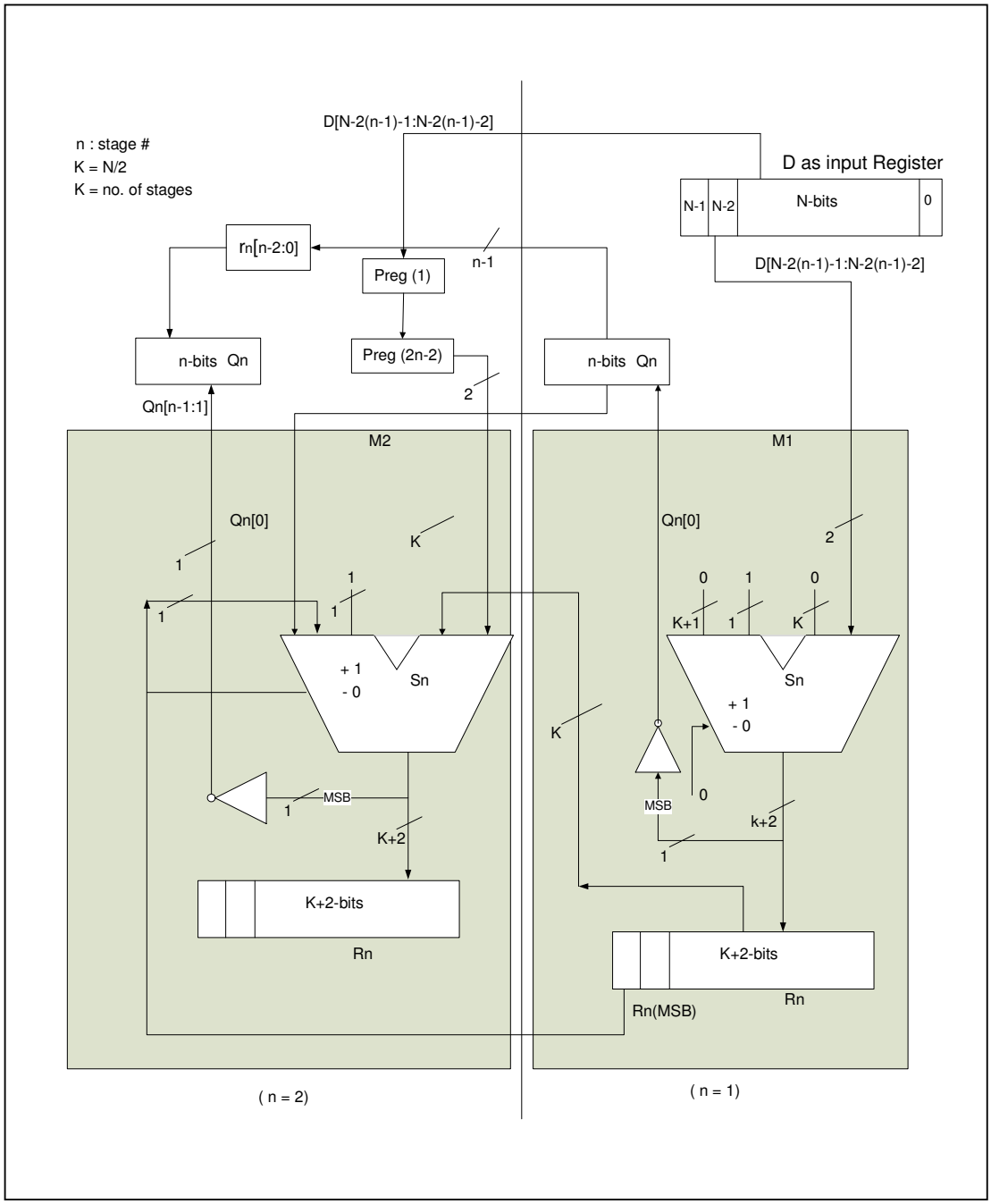


Figure.4.7 The details of the M1and M2 modules and their inter-connections.

Table-4.1 Throughput of non-pipelined (NP) and the proposed pipelined (PL) architecture.

	Time per square root evaluation (ns)	
Number of operations	NP	PL
1.00E+00	1.60E+02	1.25E+02
1.00E+01	1.60E+02	1.56E+01
1.00E+02	1.60E+02	4.71E+00
1.00E+03	1.60E+02	3.62E+00
1.00E+04	1.60E+02	3.51E+00
1.00E+05	1.60E+02	3.50E+00
1.00E+06	1.60E+02	3.50E+00
1.00E+07	1.60E+02	3.50E+00

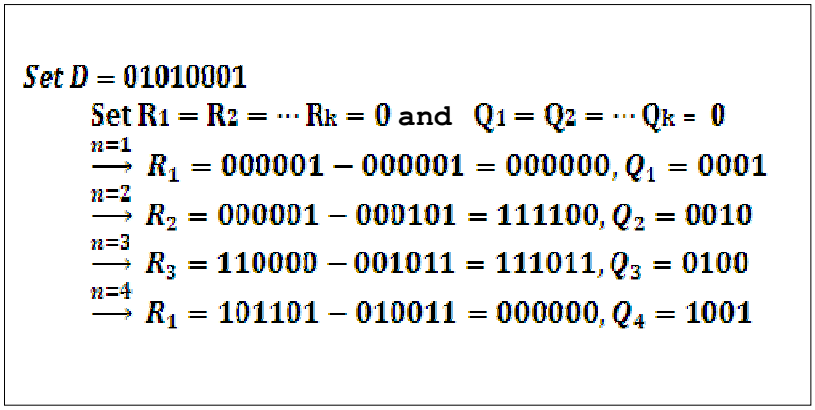


Figure-4.8 Dry run for an example using a 4-stage pipelined system (8-bit radicand).

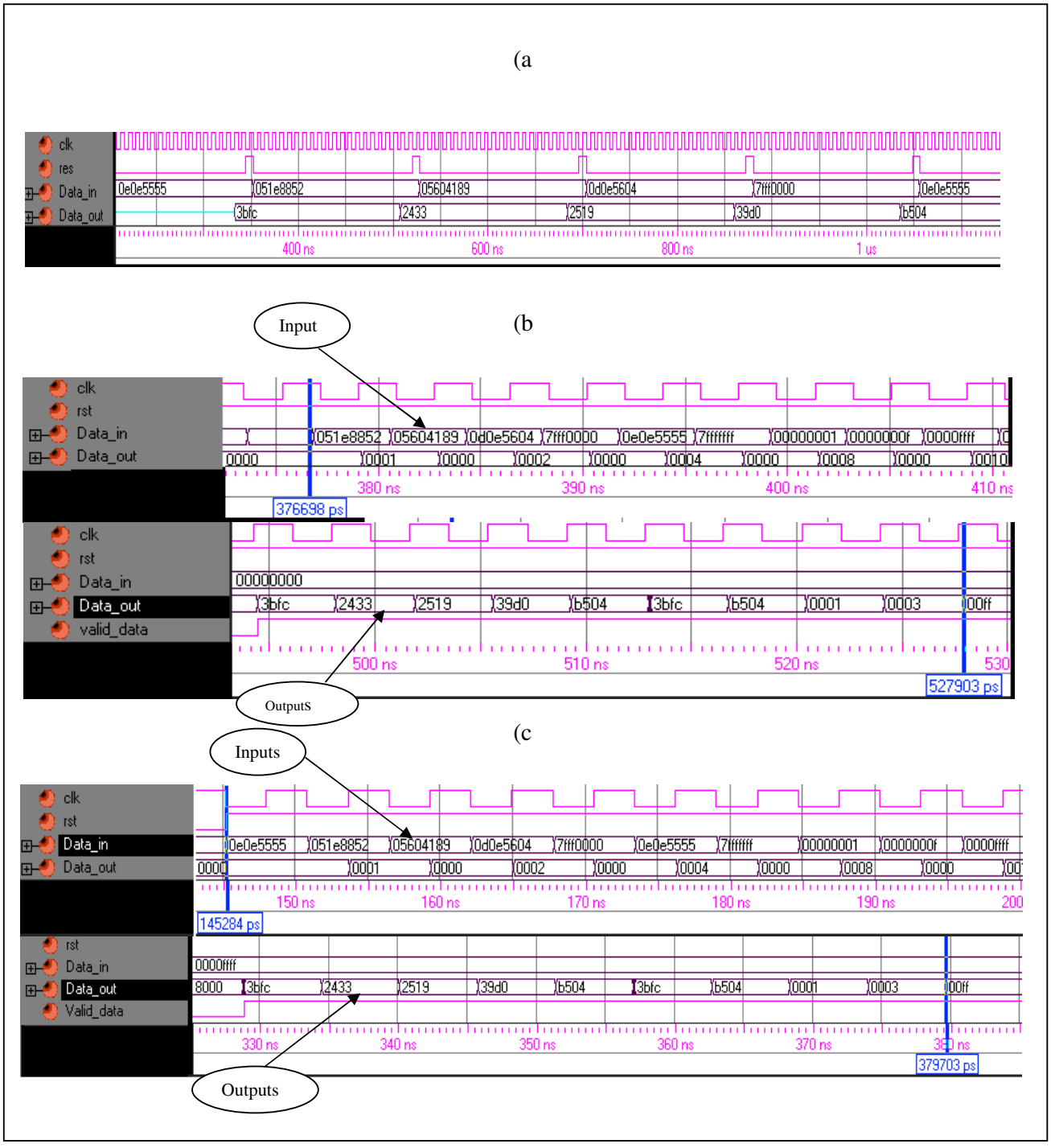


Figure-4.9 Post layout simulation of the proposed algorithm (a) using a non-pipelined architecture operating at 8 ns on the Virtex2Pro (b) using a pipelined architecture operating at approximately 3.5 ns on the Virtex2Pro (c) using a pipelined architecture at 5.63 ns on the Spartan3E.

Million floating-point operations per second per slice (MFLOPS/slice) is another relevant metric used to evaluate FPGA based systems. It is pertinent to mention that the power required by a slice of an FPGA is defined by its operating frequency (Shang 20020. Thus, the time and power consumption of a system are inter-related for the same FPGA. A benchmark is required which can provide time and power measures simultaneously.

Table-4.2 illustrates a comparison based on power, execution time and number of stages engaged by an architecture reported in the literature (Li 1997, Promsopa 2001, Ronald 2004, Thakkar 2006-a, Wang 2003). For the proposed system, the power consumption was calculated using the Xpower utility (Xilinx 2010), whereas the time required for the same is measured from the post layout simulation. The last column of Table-4.2 was evaluated by using our proposed benchmark. The data in the fifth column represents the expression $C*S*F$, where S is the number of slices, F is the maximum operating frequency, and C= 0.001475 or 0.001629 is a constant value related to power and deduced from the device data sheet for the Virtex2 and Spartan3E, respectively. The examination of Table-2 reveals that our proposed system produces the minimum value for $C*S*F$ among all square root solutions. Furthermore, in comparison to (Thakkar 2006-a) the great advantage of our proposed pipelined system is that it yields a four-fold speedup; the 28-stage implementation of the former shows a 25% improved C*S*F product whereas its 59-stage solution almost doubles the consumption in comparison to our solution.

Table-4.2. Performance comparison based on the power consumption and time for fixed and floating-point based systems and various devices (TESQR: Time to evaluate the square root).

.

Architecture	Stages	Implementation	TESQR(ns)	C*S*F (mw)	Device	Slices/LE	mW*ns
32-bit non-restoring fixed-point [12]	-	Non-pipelined	1170	65.48	Altera 10K20RC240-4	161 (LE)	76611.60
32-bit modified non-restoring fixed-point	-	Non-pipelined	160	11.78	XC2VP2-7	63	1885.58
Single Precision [28]	-	Pipelined	168	1309.19	XC2V1000	1313	219944.30
Single-precision [4]	15-Stages	Pipelined	1.8	6389.70	XC2V4000	4332	11501.46
Double-precision [18]	28-Stages	Pipelined	14.47	152.18	XC2V6000	1433	2202.11
Double-precision [14]	47-Stages	Pipelined	7	364.90	XC2VP2-7	1730	2554.30
Double-precision [18]	59-Stages	Pipelined	7.88	529.59	XC2V6000	2700	4173.19
Fixed-point Modular array, 32-bits [30] [31]	16-Stages	Pipelined	101	30.37	XCV100E-8	312	3067.37
Proposed fixed-point 32-bit	16-Stages	Pipelined	3.50	270.86	Virtex2 XC2VP2-7	709	1002.17
Proposed fixed-point 32-bit	16-Stages	Pipelined	5.63	203	Spartan3E XC3S1600E-5	704	1142.89

4.3 Error Analysis of Fixed Point Householder

As per our best knowledge HH was never implemented on FPGA. It would be therefore, very pertinent to evaluate its error bound and compared it with report CJA error. To define error limits, an error analysis methodology discussed in (Ortega 1963, Wilkinson 1962), is used. Such a technique provides an error bound for an algorithm by considering perturbation in the evaluated data (APU 2007) whereas, a perturbation which is observed due to the rounding effect of an iterative process, depends upon the word-length of the machine (Ortega 1963, Wilkinson 1962). To develop an error bound relationship for fixed point implementation of HH on FPGA, let λ_j be the j^{th} analytical eigenvalue whereas $\bar{\lambda}_{jf}$ and $\bar{\lambda}_{jr}$ represent the same evaluated by employing floating and fixed point architectures, respectively. Then an estimate of error bound for j^{th} eigenvalue can be expressed as:

$$\frac{\left|\lambda_j - \bar{\lambda}_{jr}\right|}{\bar{\lambda}_{jr}} \quad where \quad j = 1,2,...., k \qquad \text{…….. Eq (4.1)}$$

By using triangular inequality, we can write

$$\frac{\left|\lambda_j - \bar{\lambda}_{jr}\right|}{\bar{\lambda}_{jr}} \le \frac{\left|\lambda_j - \bar{\lambda}_{jf}\right|}{\bar{\lambda}_{jr}} + \frac{\left|\bar{\lambda}_{jf} - \bar{\lambda}_{jr}\right|}{\bar{\lambda}_{jr}} \quad OR \quad \varepsilon \le \varepsilon_1 + \varepsilon_2 \qquad \text{...……...Eq (4.2)}$$

where

$$\varepsilon \equiv \frac{\left|\lambda_j - \bar{\lambda}_{jr}\right|}{\bar{\lambda}_{jr}}, \quad \varepsilon_1 \equiv \frac{\left|\lambda_j - \bar{\lambda}_{jf}\right|}{\bar{\lambda}_{jr}}, \quad \varepsilon_2 \equiv \frac{\left|\bar{\lambda}_{jf} - \bar{\lambda}_{jr}\right|}{\bar{\lambda}_{jr}}. \qquad \text{……. Eq(4.3)}$$

Error analysis based on observations that $\bar{A}_1 = A_1, \bar{A}_2, ... \bar{A}_{k-1}$ is a sequence of matrices computed by a machine having *m* word-length (Ortega 1963). Assuming that a matrix A_i is computed from the same machine, such that A^{i+1} and $\bar{A}_{i+1}$ are its tridiagonal reduction matrices evaluated by using analytical and numerical techniques

respectively. And also if $\lambda_1^{(i)} \geq ..... \geq \lambda_k^{(i)}$ and $\overline{\lambda}_1^{(i)} \geq ..... \geq \overline{\lambda}_k^{(i)}$ are the Eigen values of A^{i+1} and $\overline{A}_{i+1}$ respectively, as in (Soderquist 1997), the perturbation theorem provides maximum rounding effect in j^{th} eigenvalue at $(i+1)^{\text{th}}$ iteration as follows:

$$\max_j \left| \lambda_j^{(i+1)} - \overline{\lambda}_{jf}^{(i+1)} \right| \leq \left\| X_i \right\|, \qquad \text{..... Eq (4.4)}$$

where $X_i = A_{(i+1)} - \overline{A}^{(i+1)}$ for $i = 1,2,...,k-2$. Substituting Eq. (4) into Eq. (3), ε_1 can be written as:

$$\varepsilon_1 = \max_j \frac{\left| \lambda_j^{(i+1)} - \overline{\lambda}_{jf}^{(i+1)} \right|}{\overline{\lambda}_{jf}^{i+1}} = \frac{\sum_{i=1}^{N-2} \left\| X_i \right\|}{\overline{\lambda}_{jf}^{i+1}} \qquad \text{..... Eq (4.5)}$$

4.4 Comparison with Jacobi Based CORDIC Algorithm

The proposed architectures have been tested by using five input data sets of images at different resolutions. The performance of the architectures have been evaluated in the context of accuracy, time of execution and power consumption and then compared with CJA-based system.

a) Accuracy :

To calculate the accuracy of the proposed architectures, as discussed in Section-3, an experimental set up has been created on ML403 board with XC4VFX12 device. Five data sets, with twenty images in each set, were formulated to generate co-variance matrices. These co-variance matrices were distinct because they were generated using images of different resolutions (Spacek 2008).

Table- 4.3 gives a comparison of first leading Eigen values calculated by using the proposed architectures and Matlab *eig* function for images of different resolutions. It

is evident from the table that the floating point architecture has precision equal to that of Matlab except for the image at 300 x 233 resolution, where a nominal error is observed. The last column of Table- 4.3 shows fixed point architecture error compared to Matlab. Although there is a small loss in accuracy, which could be ignored for a pattern recognition application (Siwabessy 1999) yet it is preferred because the architecture is inherently faster compared to floating point and PC based evaluation. For a 20 x 20 input matrix the observed error of fixed point architecture was 0.021% whereas, for a 16 x 16 matrix the reported error for CJA was 0.43% (Bravo 2006). Thus, our fixed point architecture is considerably accurate than CJA.

Error bound of an algorithm is evaluated by considering perturbation in the evaluated data whereas, a perturbation which is observed due to the rounding effect of an iterative process, depends upon the word-length of a machine (Wilkinson 1962, Ortega 1963). If N and L represent maximum allowable iteration and machine word length respectively then according to Ortega the error bound of HH (ε_{HHM}) is

$$\varepsilon_{HHM} \leq \left[\frac{(6.01N^{3/2} + 286N - 596)m}{1 - (6.01N^{3/2} + 286N - 596)m} \right] * \|A_1\|. \qquad \ldots\ldots \text{Eq (4.6)}$$

where $\|A_1\|$ represents the matrix norm of a co-variance matrix A_1. This norm could be either the Euclidian or $\|.\|_2$. Table- 4.4 is computed using Eq. (6) for N=20, $m=2^L$, L=32 and Euclidian-norm of $\|A_1\|$.

Paul *et al.* have evaluated the error bound of CJA (ε_{CJA}) for its fixed point implementation by using Wilkinson's work given by (Wilkinson 1962, Paul 1995).

$$\varepsilon_{CJA} \leq \frac{1}{\sqrt{2}} C(2N-4)^{1/2} \frac{(N-1)N}{2} 2^{(-d-1)} \qquad \ldots\ldots \text{Eq (4.7)}$$

Table-4.3 Percentage error of proposed architectures with reference to Matlab based eigenvalues.

Image resolutions	First Leading Eigenvalue			% Error	
	Matlab	Float	Fixed	Float	Fixed
112x87	2977474304	2977474304	2976270848	0	0.0404187
118x 146	8385567744	8385567744	8386328576	0	0.0090731
263x 204	16393589760	16393589760	16394962944	0	0.0083756
300x 233	21360439296	21360447488	21351821312	3.835E-05	0.0403455
376x 292	33547739136	33547739136	33550653440	0	0.008687

Table-4.4 Error bound of HHM for five image data sets using fixed-point architecture shown in Figure-4.2.

Image Dimension	Error
122 x 87	7.20E-11
188 x 146	1.20E-10
263 x 204	1.69E-10
300 x 233	1.93E-10
376 x 292	2.41E-10

In Eq. (2) $2^{(-d-1)}$ is machine accuracy where d is digits length according to machine architecture. Taking N =20, C=5 and d=21 for a 64-bits machine architecture, the error bound of CJA, using Eq. (7), is evaluated as 6.36E-3. The worst error bound in Table- 4.4 is 2.41E-10 which is significantly better than 6.36E-6. This demonstrates that fixed point HH implementation, as discussed in Section-3 is better than CJA.

b) Power Consumption :

Power consumption by an FPGA is dependent upon the resources engaged by an architecture and its operating frequency (Shang 2002). To calculate the power dissipation of our architecture, experiments were performed on ML403 Xilinx board having Virtex4Fx device which has 5472 slices, 1094 Flip-flops, and 32 DSP units.

The hard processor core of the FPGA was instantiated at 110 MHz. It was observed that floating architecture (Figure-4.1) occupied 1915 slices while fixed with 31 rounding bits consumed 1970 slices for a 20 x 20 image data matrix.

CJA implementation on Virtex II-Pro (XC2VP7) with 24 rounding bits for an 8 x 8 image data matrix consumed 2512 slices. The slices used by proposed architectures were less than CJA, but both systems were using different Xilinx devices. It was investigated that XC2VP7 Xilinx device had 4928 slices containing 11088 logic cells. Whereas, XC4VFX12 Xilinx device had 5472 slices containing 12312 logic cells (Daempling 2010, Xilinx 2010). Thus number of logic cells per slice are constant for both Xilinx devices.

The power consumed by CJA was then calculated and it was ~408 mw at 110 MHz operating frequency (Shang 2002). Whereas, power consumed by our floating and fixed architectures was ~283 mw and ~291 mw respectively as shown in Table- 4.5.

Table-4.5 Power and time estimation of floating and fixed architectures on Virtex4Fx.

Performance Parameters		Architectures	
		Floating	Fixed
Resources	Slices	1915	1970
	LUTs %	18	18.007
	Flipflops %	21	21.006
	Power dissipation (m watts)	283	291
Execution Time	Software (µsec)	29.46	13.14
	Hardware (nsec)	10	170
	Interconnects	3.6	3.6
	Total (µsec)	33.07	16.91

In short, keeping in view the size of our data set and number of slices engaged by our architectures, one can comfortably claim that our proposed architectures are resource efficient compared to CJA.

c) Execution Time :

Execution time of HH consists of time consumed by a) software layers; b) hardware layers and c) interconnects. To calculate execution time of software layers Xilinx microprocessor debugger (XMD) tool was used. XMD provides low level machine instructions and the time required to execute these instructions can be calculated using PowerPC data sheets (IBM 2006, PPC 2005). It is observed that fixed point and floating architectures consumed 1446 and 3274 cycles with 13.14 µs and 29.46 µs respectively. To calculate time for hardware layers, post layout synthesis report was used which gave 10 ns and 170 ns for floating and fixed point architectures respectively.

In addition to hardware and software layers time, interconnects between different components in the co-design architecture also consume time which cannot be overlooked. Although these interconnects are implemented through hardware-based FIFOs, yet they consume ~9 ns for 32-bits data. And for a 20 x 20 matrix the time consumed by the interconnects is ~3.6 µs. Thus, the total execution time for fixed and floating architectures are ~16.91 µs and ~33.07 µs respectively as shown in Table-4.5.

Bravo *et al.* demonstrated that CJA co-design implementation consumed 24.15µs for a 16 x 16 matrix (Bravo 2006). Our fixed point architecture, which is also based on co-design technology, consumed 16.91 µs for a 20 x 20 matrix. Comparison of the

data revealed that our proposed fixed point architecture is at least 30% time efficient than CJA.

Based on the results summarized in Table- 4.3, 4.4 & 4.5, one can conclude that fixed architecture, shown in Figure-4.2, provides a better implementation for the evaluation of Eigen values on FPGA.

4.5 Summary

Fixed three-layer architecture for the implementation of Householder algorithm on FPGA was presented. In most of the digital signal and image processing applications, Eigen values are integral part but computationally intensive. Recently, coordinate rotation digital computer (CORDIC) based Jacobi algorithm (CJA) has been used for the evaluation of Eigen values on FPGA. CORDIC uses trigonometric functions, counterpart of square root, for the evaluation of eigenvalues. However, CJA inherits some error in the eigenvalue's evaluation even for a small symmetric matrix because CORDIC computes trigonometric functions by using vector rotation approximation. In this chapter, we present a new FPGA-based architecture to implement Householder (HH) method for the evaluation of Eigen values by employing non-restoring square root algorithm.

The validity of the proposed architecture has been checked by using input data of images at different resolutions. Eigen values of the input matrices have been evaluated. The performance of the proposed architecture was gauged by comparing accuracy time consumed and power dissipation with its counterpart.

Our proposed architecture demonstrated an improvement up to 30% in time and 10^{-7} decimal places in accuracy than CJA. The validity of the proposed architecture is checked by using five data sets for a real-time face recognition system. It could be a better choice for the Eigen values based face recognition system.

CHAPTER 5

EFFICIENT EVALUATION OF VECTOR NORMALIZATION

5.1 Introduction

Eigen values are used for the extraction of features in a face recognition system. Their evaluation is time consuming but is an important part of a pattern recognition system (Niklas 2007, Stavros 2003). Decomposition and orthogonalization are the two main strategies to compute Eigen values. Orthogonalization is faster compared to decomposition (Sharma 2007) and it also provides better accuracy. However, it has convergence problem when the co-variance matrix is associated with high-resolution images. Modified Gram-Schmidt orthogonalization (MGSO) used in fast PCA solves the convergence issues but requires additional vector normalization (Sajid 2008-a). Normalization is an essential but time consuming part of MGSO and is applied iteratively.

Normalization in addition to other arithmetic operations involves square root operations. A fixed-point square root operation is efficient than its floating-point counterpart, but conventionally it loses precision due to algorithmic complexity. For the sake of efficiency Piromsopa *et al*. have implemented non-restoring algorithms on FPGAs to meet stringent time requirements of a real time system (Piromsopa 2001).

In face recognition Euclidian length normalization may provide more accurate results than norm2 (Sajid 2008-a, He 2005). Its time complexity is $O(n \log n)$, where n is the rank of a square matrix (Stavros 2003, Gavish 2006).

Normalization could be made more time efficient using a co-design process (Chin-Chin 1996, Benkrid 2010). Co-design methodologies are employed to achieve trade-offs among execution time and power consumption. In this chapter, three co-design architectures are presented by employing fixed-point operations, pipelining and instruction level parallelism (ILP) for vector and matrix normalization on FPGAs. Floating point operations are costly in hardware, especially the square root, since they use more chip resources and consume more cycles compared to integer operations (Sajid 2010-a, Oberstar 2007, Liao 2000, Yamin 1997, Peter 1996). Integer arithmetic by its very nature is resource and time efficient and its precision depends upon the chosen *q.n* format (Wilkinson 1962, Ortega 1963, Burden 1997). To select appropriately a *q.n* format, a debugger is employed, which provides an estimation of the operand ranges for various types of operations.

5.2 Design Flow

In Figure 5.1, a design flow of vector normalization is presented at abstract level. Step-1 of the design flow shows a block containing a normalization algorithm with input and output connectors. In this step a high level algorithmic verification is completed. Step-2 provides a low level detail of the algorithm including arithmetic operations, data structures and data movement. To ensure optimum utilization of the available resources, co-design methodology is normally used for the implementation of an algorithm, which is discussed in Step-3. In this step, time consuming parts of the algorithm are identified. Loop unrolling and pipelining of computationally intensive parts can lead to time efficient implementation, but may increase power dissipation. In Step-4 a trade-off between execution time and power dissipation is finalized by trying various combinations of hardware parallelism.

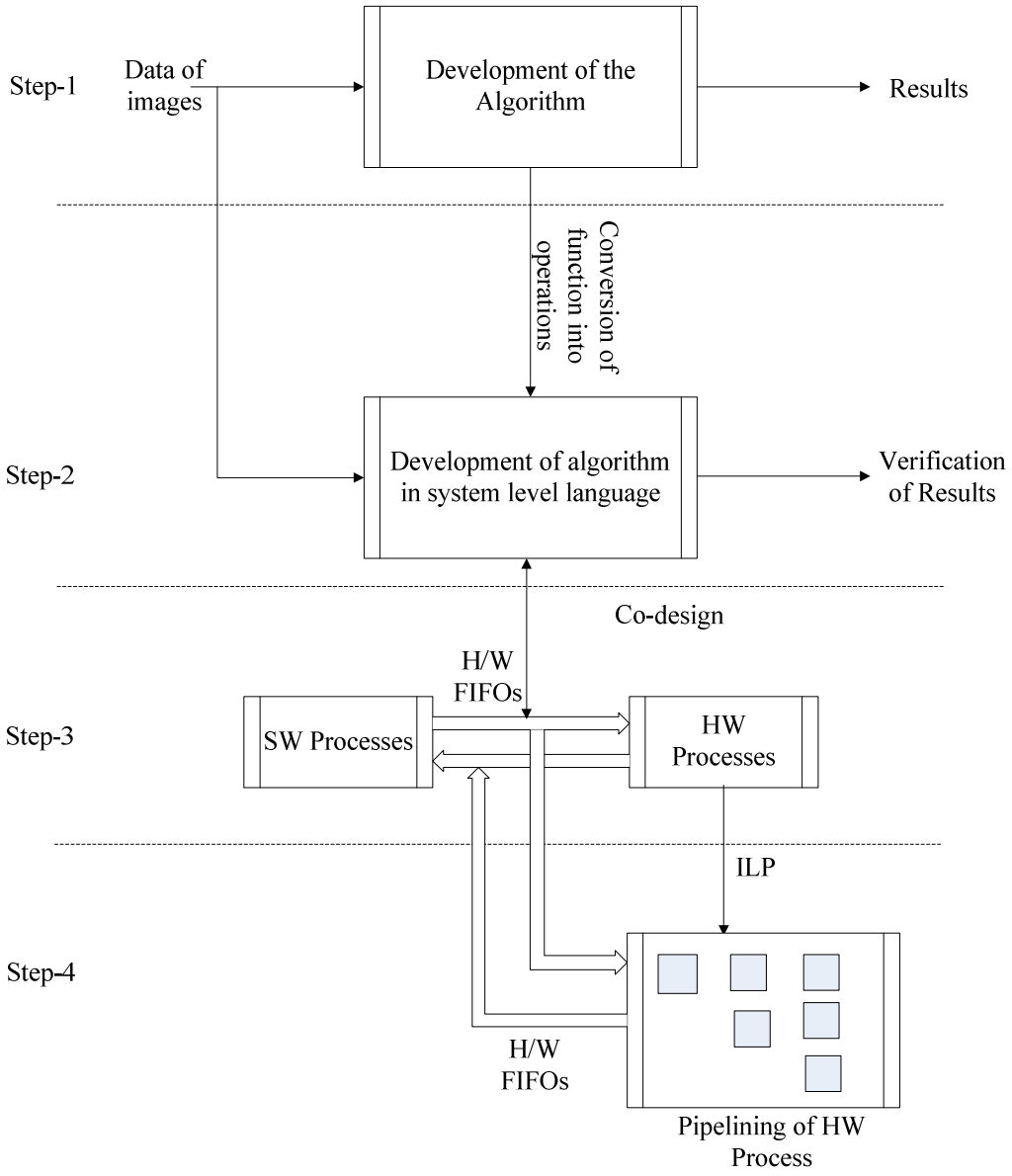


Figure-5.1 Illustrate implementation of an algorithm using co-design methodology.

5.3 Architectures

5.3.1 Embedded Processor Based— Archi-I

Figure 5.2 shows three layers architecture for vector normalization on an FPGA. The application layer consists of a data producer (DP), a data consumer (DC) and a fixed-point normalization unit (FxN). DP and DC are responsible of providing data to FxN block and getting it back from real-time operating system (RTOS) layer. FxN translates floating-point instructions into fixed-point integer instructions. In order to facilitate these interactions, two FIFO queues are involved. The hardware layer does not contain floating-point unit (FPU) because it targets fixed-point operations created by FxN. FPU has been avoided because it requires high FPGA resources and involve large number of machine cycles per operation. The architecture shown in Figure-5.2 computes normalization without floating instructions. It is assumed that the proposed architecture should be time efficient than a software-based floating architecture (Sajid 2010-a). Because, a pure software based floating-point implementation of an algorithm is much slower compared to its fixed-point hardware realization, therefore, an efficient hardware implementation of FxN unit is our primary objective.

5.3.2 Dedicated Component Based — Archi-II

Figure 5.3 presents our second architecture referred to as dedicated component based architecture. In this architecture, FxN block of Figure 5.2, was redesigned and labeled as IPNM. It was then integrated into the processor local bus (PLB) of the PowerPC core processor as IPFxN instead of integrating it with the On-chip peripheral bus (OPB) of the previous architecture. PLB is much faster than OPB because it is a native bus to the processor and operates at the master clock speed (IBM 2001).

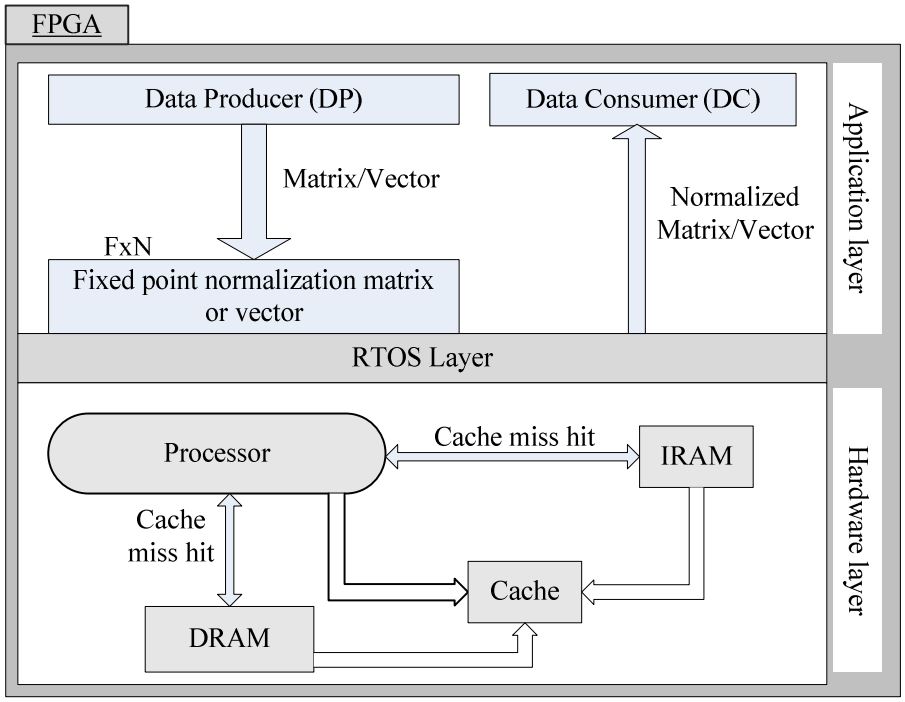


Figure-5.2 A three layer architecture to achieve fixed-point vector normalization.

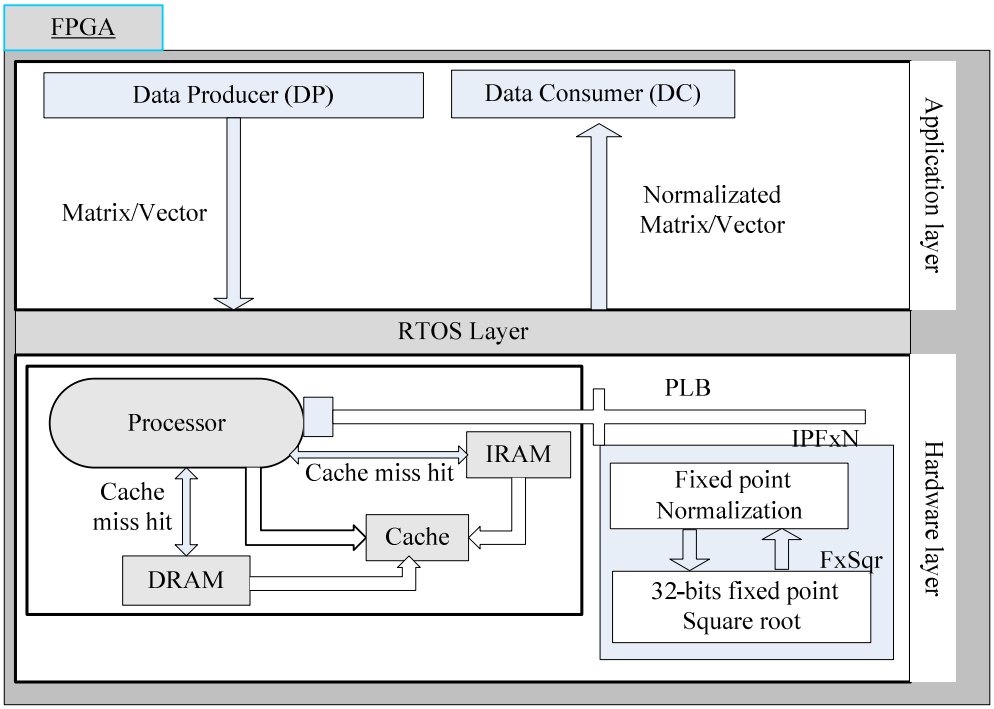


Figure-5.3 A three layer FPGA-based architecture for fixed point vector normalization.

IPFxN performs multiplication and add/subtraction operations similar to MAC (multiply-and-accumulate) operations. In addition to the MAC operations, division and square root operations are also required. The fixed-point square root (FxSqr) operation is not advisable to be implemented within IPFxN because it is a computationally intensive unit.

On the other hand, nested hardware processes require stack memory for the call back function pointer. The alternative is to design FxSqr as a separate process and connect it through FIFOs with main component of IPFxN, as shown in Figure 5.3. This will remove the need of the stack memory and its associated controller. The stack memory may be a less expensive solution, but its management requires customization of the embedded core's functionality.

Thus, in this architecture, fixed-point normalization module is moved from the software layer (Figure 5.2) to the hardware layer as shown in Figure 5.3. This has increased slightly the consumed resource of the target FPGA. But, keeping in view the design strategy, it is assumed that it could significantly improve the execution time. The resources consumed by the hardware layer of Figure-5.3 could be reduced further by appropriately eliminating the processor core from the hardware layer.

5.3.3 PC-Based Implementation— Archi-III

Figure 5.4 presents our PC-based FPGA architecture without PowerPC embedded processor core. In this architecture, only the computational unit IPFxN is placed inside the FPGA device. The DC and DP are implemented in a C shell on the host machine and the PCI bus acts as the interface between DC and the FPGA. The execution time can be estimated using the following Equation:

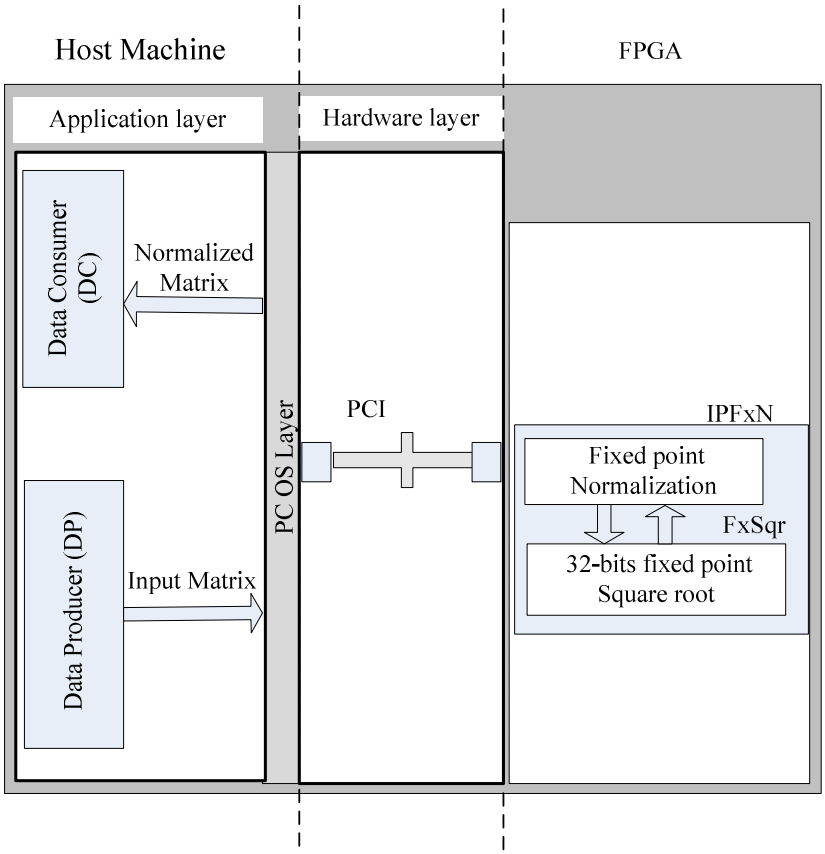


Figure-5.4 PCI-based architecture for vector normalization wherein the data is provided by the host machine to a 32-bit fixed-point normalization unit defined in an FPGA.

$$t_{ex} = t_H + t_{if} + t_p + t_{IP} \quad \text{.....Eq (5.1)}$$

Where t_{ex} is the execution time for normalization of a matrix or a vector, t_H is the time consumed by the host application, t_{if} is the time taken by the two ends of the hardware layer for data transfers, t_p is the time consumed on the PCI bus and t_{IP} is the time taken by the IP core. The IP core has the same design as in Figure 5.3. The two ends in the hardware layer are: (a) the FPGA on the Annapolis Microsystems Wildstar II-PCI board with two Xilinx Virtex II XC2V6000-5 (b) the software controller at the PC OS side.

The PCI bus operates at 133 MHz and can send 32-bit data in 8 ns for a matrix of rank 20 (Wildstar-II). But usually the PCI bus takes more time due to some acknowledgement signals and possible packet loss overheads. It has been observed that t_p is about 6.4 µs for a square matrix of rank 20. The value of t_H dominates in Equation (5.1) for this architecture because the respective process executes in a C shell on the PC. This architecture uses minimum resources of the target FPGA because it does not require FIFOs and embedded core. The share of t_H value in Equation (5.1) will be reduced when number of normalization processes are in millions.

5.4 System Performance and Comparative Analysis

The precision of fixed-point normalization depends on the proper adjustment of decimal point operands/results in *q.n* format. Thus, the accuracy of normalization is dependent upon input data; however, it does not depend upon the implementation methods discussed in Section-5.3. To estimate the performance of the proposed architectures for a face recognition system, four popular databases have been selected

namely: Yale, ORL, FERET and CAS-PEAL databases (Yale2002, Face 2009, Wen 2008, Phillips 2000).

Yale faces database contains 165 images of 15 subjects, with 11 images per subject. These images have variation in illumination, including center light, left light and right light as shown in Figure 5.5. ORL database contains 400 images of 40 personnel with variations in facial expression, like open and closed eyes, smiling and non-smiling, with and without glasses as shown in Figure 5.6.

Similarly CAS-PEAL is a Chinese faces database containing 99,594 images of 1040 subjects. A total of nine cameras were used simultaneously to capture images across different poses, facial expressions, lighting and angles as shown in Figure 5.7. Finally FERET is a big faces database containing 14,126 images of 1199 subjects and a specimen from those is shown in Figure 5.8.

The performance of the proposed architectures is estimated by evaluating accuracy, execution time and power consumption. A set of twenty randomly selected images from the four databases were tested. The images were selected having different resolutions i.e., 70x80, 128x192, 96x120 and 100x100 for the ORL, FERET, CAS-PEAL and Yale databases respectively.

To measure the accuracy RMS metric is used. As envisaged, the same accuracy was observed from the three proposed architectures, discussed in Section-5.3. Because error in fixed-point depends upon *q.n* format instead of implementation techniques. For reference, floating-point based normalization was performed using Matlab and it was then compared with the evaluated values obtained from the proposed architectures. The minimum and maximum RMS errors are shown in Figure 5.9.

Figure-5.5 Sample images from Yale faces database.

Figure-5.6 Sample face images from ORL database.

Figure-5.7 Sample images from CAS-PEAL faces

Figure-5.8 Sample images from FERET faces database.

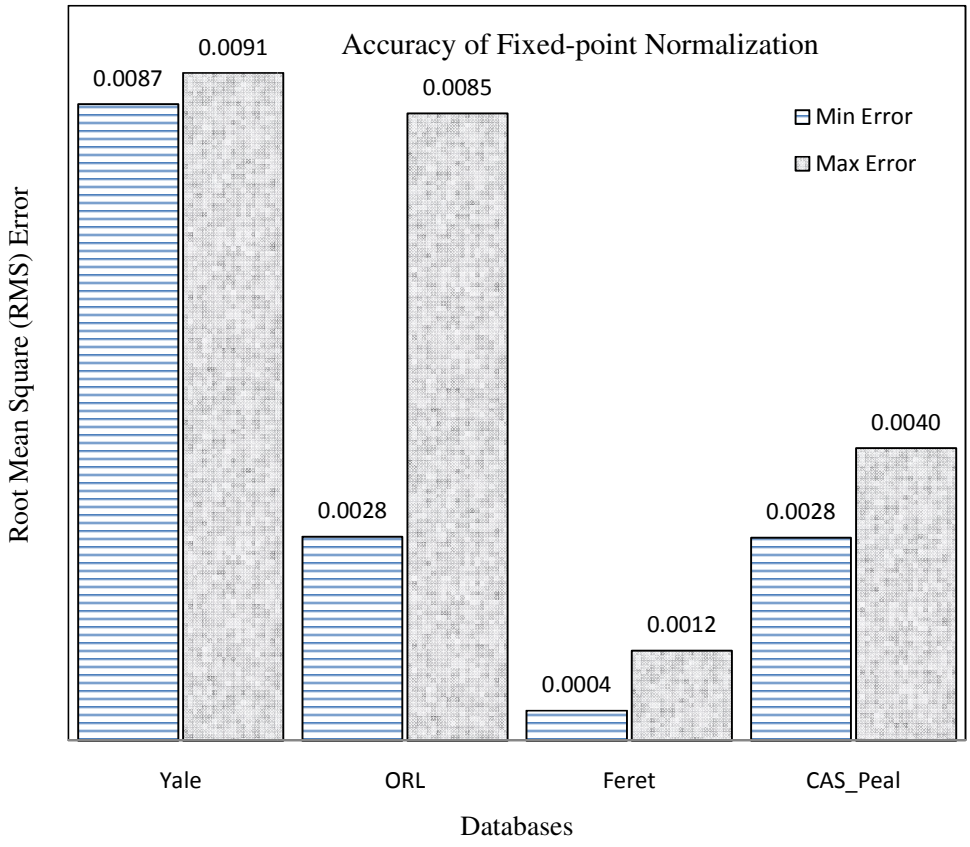


Figure-5.9 The minimum and maximum observed errors of four databases w.r.t. floating point values.

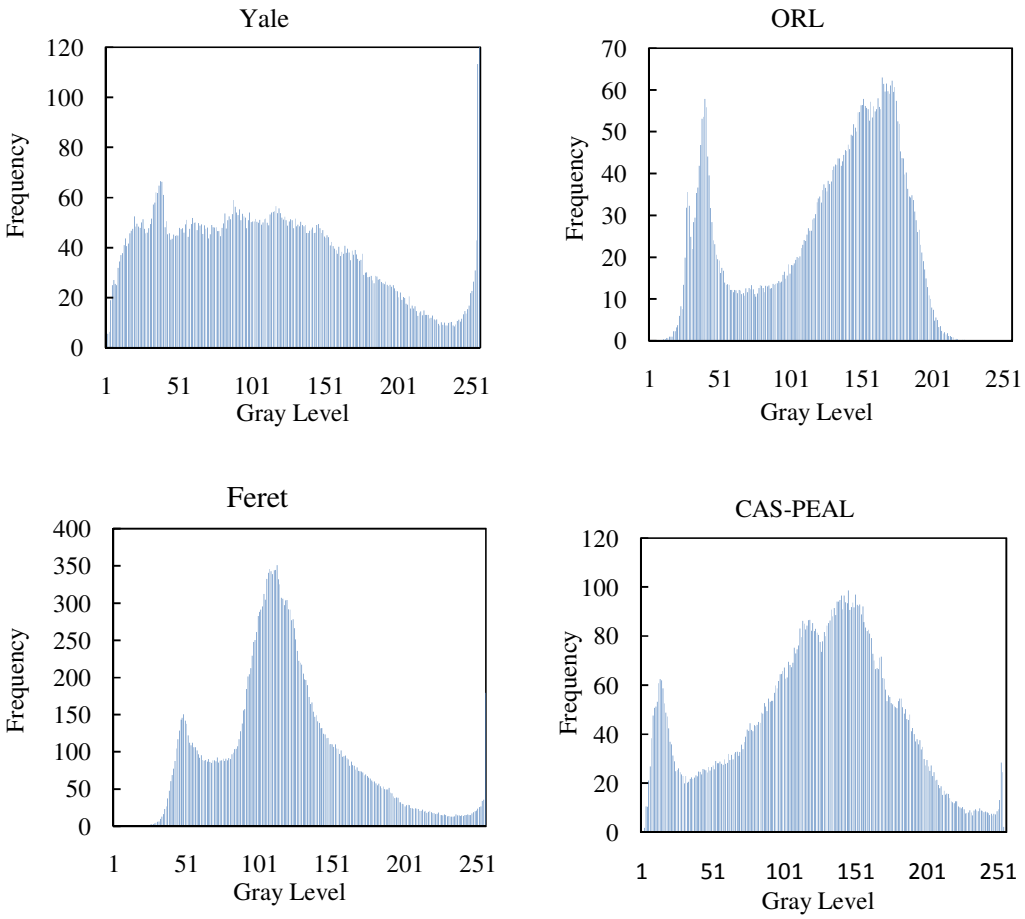


Figure-5.10 Average histogram of twenty images from four face databases showing frequency of gray levels.

The average histogram of gray levels for randomly selected images from the databases under discussion is shown in Figure 5.10. The highest frequency, as evident from Figure-5.10, is observed in FERET whereas, it is lowest in Yale. On the other hand, ORL and Yale have the smallest distribution of values in the central region than FERET and CAS-PEAL. Furthermore, their maximum frequency in central region is smaller than FERET and CAS-PEAL. On the other hand, the maximum frequency of gray levels in central region of ORL and Yale is approximately 60 and 50 respectively. Whereas, it is 300 and 90 for FERET and CAS-PEAL respectively as shown in Figure 5.10.

In normalization process the frequency values of gray levels, observed in an image, are summed and then used as a normalization factor. Database having higher normalization factor yield relatively lower error as evident from Fig 5.9 and 5.10. However, the error range for all the databases is of the order of 10^{-3}, which is within acceptable margin for a face recognition system using MGSO (Giraud 2003).

Table-5.1 shows the execution time and power dissipation comparison of our architectures presented in Section 5.3. The architectures were evaluated based on the FPGA resources required and the time taken to complete the normalization process. The power dissipation was estimated using Xilinx data sheets and the Xilinx *XPower* tool in the integrated simulation environment (ISE) (Shang 2002, Klein 2005). The execution time of software processes was noted using the Xilinx EDK tool (EDK 2006). Whereas, the execution time of hardware processes like IPFxN was estimated using synthesis reports.

Table-5.1 represents a trade off amongst execution time, resource utilization and power dissipation for three architectures discussed in Sections 5.3.1, 5.3.2 and 5.3.3. Archi-1 cannot be materialized using piplining and ILP techniques because it uses a computing unit (FxN) which is implemented in the operating system shell. Whereas, Archi-II and Archi-III were tested for three possible implementations: a) pipelined; b) un-pipelined; and c) loop unrolling.

The architecture employing pipelining and loop unrolling (PLU) techniques is usually time efficient. It is observed that Archi-I, which is implemented without PLU, consumes 232.14 µs, whereas Archi-II with PLU takes 9.3 µs. But Archi-III with PLU takes 15073 µs, out of those 15064 µs are for data feeding. Thus, execution takes only 9 µs. It is pertinent that execution time of Archi-III for large number of operation, as shown in Table 5.1 is significantly lower than Archi-I & II. Archi-III is 200 times faster than archi-I when 100 million operations are involved.

Normalization in time MGSO typically comprises of data feeding time and execution time. Average time in Table 5.1 is computed by using {(Ops * ET) + DFT} / Ops; where Ops is number of normalization processes, ET represents execution time for an operation and DFT is data feeding time. The data feeding time for Archi-I and Archi-II is defined by the DC and DP component of the circuit whereas the same for Archi-III is defined collectively by the PCI bus and the DC and DP units. It is pertinent to mention that the bandwidth of a PCI bus depends upon the host machine and it is much slower than the DC and DP units because it transfers data using PC1 connector. Whereas, DP/DC is transferring data by using on-chip FIFOs channels. Therefore, data feeding time of Archi-III is much higher than Archi-I&II.

Table-5.1. Trade off amongst of execution time, resource and power consumptions for three architectures on

	Performance Parameters	Architecture-I	Architecture-II			Architecture-III		
		Archi-I_WPL	Archi-II_WPL	Archi-II_PL	Archi-II_IPL	Archi-III_WPL	Archi-III_PL	Archi-III_IPL
Resources	Slices	1979.000	4587.000	4615.000	4641.000	1750.000	1765.000	1794.000
	LUTs	2266.000	7399.000	7471.000	7514.000	3402.000	3443.000	3489.000
	Flipflops	1981.000	2709.000	2710.000	2900.000	930.000	936.000	1157.000
Dissipation	Power (mw)	5.779	13.394	13.476	13.552	5.110	5.154	5.238
Input Data	Feeding Time (u sec)	346.280	346.280	346.280	346.280	15064.000	15064.000	15064.000
One Operation	Execution Time (u sec)	232.148	11.295	11.105	9.320	13.200	12.150	9.320
	Time including feeding in u sec	578.428	357.575	357.385	355.600	15077.200	15076.150	15073.320
1000 Operation	Execution Time (u sec)	2.321E+05	1.130E+04	1.111E+04	9.320E+03	1.320E+04	1.215E+04	9.320E+03
	Average Time including feeding (u sec)	232.495	11.641	11.451	9.666	28.264	27.214	24.384
One million Operation	Execution Time (u sec)	2.321E+08	1.130E+07	1.111E+07	9.320E+06	1.320E+07	1.215E+07	9.320E+06
	Average Time including feeding (u sec)	232.149	11.295	11.105	9.320	13.215	12.165	9.335
100 million Operation	Execution Time (u sec)	2.321E+10	1.130E+08	1.111E+08	9.320E+07	1.320E+08	1.215E+08	9.320E+07
	Average Time including feeding (u sec)	232.148	1.130	1.111	0.932	1.320	1.215	0.932

Archi_PL: Architecture with pipelined
Archi_WPL: Architecture without pipelined
Archi_PLU: Architecture with pipelined and loop unrolling

But this data feeding time is once for a matrix regardless of its rank. Consider a matrix of randomly selected 200 images, it requires approximately 10^6 normalization operations to complete MGSO process. Thus, for such a case, which is usually observed in image processing, there is not much difference in execution time of a normalization process by involving Archi-II&III with PLU.

It is observed that Archi-II with PLU consumes 4641silices but Archi-III also with PLU requires 1794 slices because it uses off-chip PCI connector for data transferring and FPGA resources only for normalization process. Thus, power dissipation of Archi-III is 62% less than Archi-II. By considering the data of Table-5.1, one can conclude that Archi-III with PLU provides an optimal solution for FPGA-based normalization considering the accuracy, execution time and power consumption.

5.5 Summary

Three architectures employing fixed-point hardware for the implementation of vector normalization on an FPGA have been presented. These architectures were designed and analysed on the basis of accuracy, execution time and power consumption. The impact of pipelining and instruction level parallelism was studied as well using an architecture co-design methodology. The suitability of these architectures was evaluated based on the user requirements. It is demonstrated that the host-to-PCI based architecture provides an optimum combination of accuracy, processing time and power consumption. The PCI-based architecture for an image processing application provides over 200 times faster solution compared to the software-based solution and is 62% power efficient than the IP-based architecture. Furthermore, this

architecture provides an accuracy within 10^{-3} compared with their software-driven floating point counterpart.

CHAPTER 6

FACE RECOGNITION USING GRAY LEVEL FREQUENCY DISTRIBUTION CURVE (FDC)

6.1 Introduction

Extensive research has been carried out in face recognition because of its potential use in defense, security and commercial applications (Belhumeur 1997, Zhao 2003, Huang 2009). It is a complex type of biometric due to its non-intrusive nature. Efforts have been made to mitigate the complexity of a non-intrusive environment using physiological characteristics of the human head, but with substantially reduced accuracy (Middleton 2006).

Researchers classify face recognition algorithms into three catageries: a) holistic, b) feature-based and c) hybrid (Zhao 2003). The holistic category of algorithms considers face as a global entity; feature-based algorithms treat the face as a combination of features. It is an established fact that holistic algorithms are computationally more efficient than feature-based techniques (Zhao 2003, Zhang 2005).

PCA is a classical holistic method. It computes Eigen values in $O(n^3)$ time where n is the rank of a covariance matrix (Turk 1991). Its demonstrated accuracy is less than 90% even if the system is trained with 50% images per subject (Cai 2006, He 2005).

LDA is another holistic algorithm which normally provides accuracy better than PCA. LDA works better when the dimensionality of the transformed space is one less than the classes used in training. It requires 80% of the class as training images

(Hu 2007, Eleyan 2006). Its computational cost is higher than PCA because it uses PCA along with multivariable normal distribution of a co-variance matrix.

FDA is another algorithm which works well with the reduced space and it uses LDA to optimize the sample data point projections (Belhumeur 1997, Eleyan 2006, Martinez 2001). However, both FDA and LDA use a global Euclidian structure which ignores the local face detail, thus, offering low accuracy.

LPP is a recent feature-based method that approximates the Eigen function on the Laplace Beltrami operator and preserves the features by using a neighborhood graph technique (Cai 2006, He 2005, Hu 2007). It calculates a weight matrix using an objective function which is computed for each element of the associated matrix. The neighborhood graph is a core part of the Laplacian faces but it is computationally intensive because each node has k edges, where k is the dimensionality of the reduced face space. Furthermore, the range of the neighborhood around original data points is defined by the number of Leading Eigen Values (LEV). But the selection of minimum LEVs is not addressed in LPP. At least O (n) additional time is required after space reduction using PCA. Thus, the computational cost of LPP is more than $O(n^3)$ and the maximum reported accuracy is less than 94% (He 2005).

OLPP was developed in 2006 and it has improved LPP results using an orthogonal space because orthogonality usually provides better discriminating power in face recognition (Cai 2006, He 2005, Hu 2007). On the other hand, OLPP has further increased the computational cost and provided a maximum accuracy of 96.5% for the ORL database (Cai 2006).

Kussal's work (Kussul 2006) based on permutation coding claims 99.9% recognition accuracy for ORL database. However, it uses half of the images for training and half for testing. This criterion of training and testing could not be applied for a real-time system.

In this chapter a technique has been proposed which extracts global structural information by using gray level frequency distribution curve (FDC). The proposed FDC technique needs only one training image to provide reliable recognition within $O(n^2)$ time. Ninety-eight percent accuracy has been observed with the proposed technique for ORL and PIE databases, and 96.5% for Yale database.

To improve the time efficiency, hardware realization of the proposed technique has been achieved by developing an FPGA based architecture, which provides recognition in less than 200 ns. Thus, the proposed FDC technique is a suitable candidate for reliable real-time face recognition systems.

6.2 Overview of Reduced Space Methods

Let us consider a set A of d sample/training images having n^2 pixels each: assume that each image x_i in $A = \{x_1, x_2, \ldots, x_d\}$, *for* $i=1, 2, \ldots, d$, belongs to one of m classes $\{X_1, X_2, \ldots, X_m\}$. Let us also consider a linear transformation mapping of an original $d*n^2$-dimensional image space of A into a d-dimensional feature space as discussed below.

6.2.1 Eigen Faces

In this technique the basis vectors are computed in the reduced dimensional space. They 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.

The co-variance matrix is computed as $C = A^T A$, and the basis vectors for *d*d* dimensions are computed instead of original space $n^2 * n^2$, as

$$t_j = \frac{Au_j}{\left\| Au_j \right\|} \; and \; then \; T_{n^2 d} = \left[t_1 \, t_2 \ldots\ldots\ldots t_d \right]$$

Then, we collect the projection of each of the *h* leading Eigen vectors on each of A_i in a matrix $W_{d \times h}$ as follows:

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

.

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:

$$D_N = N - J \qquad \ldots\ldots$$

Eq (6.1)

where *J* is the mean of the *A* matrix. The following equation represents the projected space and Equation (6.3) represents an error for an input image in the training image space.

$$W_N(j) = (D_N)^T t_j ; \quad j = 1, \; 2, \; \ldots\ldots, \; h \qquad \ldots\ldots \text{ Eq (6.2)}$$

$$V_N(i) = \left\| W_N - (i^{th} column \; of \; W^T) \right\| ; \quad i = 1, 2, 3, \ldots\ldots, d. \qquad \ldots\ldots \text{ Eq (6.3)}$$

PCA arranges the given data into two classes and then PCA maximizes the variance for the two classes but it ignores local details in computing the variance. However, it has the minimum time complexity $O(n^3)$ for computing the basis vectors.

6.2.2 Fisher Faces

Eigen faces maximize the class variance and ignore within class scattering. This limitation has been removed in LDA. Fisher faces use LDA to seek the direction of an efficient discrimination between classes (Hu 2007). LDA computes a variance between classes and a variance within class using the original sample space (He 2005, Eleyan 2006). Fisher faces usually performs better when the data of a class is unimodal (Hu 2007).

Let V_b and V_w be the variance between classes and within a class, respectively:

$$V_b = \sum_{i=1}^{m} (u_i - u)\ (u_i - u)^T \quad \ldots\ldots \text{ Eq (6.4)}$$

$$V_w = \sum_{i=1}^{m} \sum_{j=1}^{N_j} \left(X_j^i - u_i\right)\ \left(X_j^i - u_i\right)^T \quad \ldots\ldots \text{ Eq (6.5)}$$

where N_j is the number of images in a class m is the total number of classes and X_j^i is the j^{th} sample in the i^{th} class. An optimal projection W_{LDA} to maximize the ratio of the determinant $\frac{V_b}{V_w}$ is:

$$W_{LDA} = \arg \max_W \frac{|W^T V_b W|}{|W^T V_w W|} \quad \ldots\ldots \text{ Eq (6.6)}$$

$W_{LDA} = [W_1, W_2, \ldots, W_h]$, W_{LDA} contains the h largest Eigen vectors. Projections between classes can be deduced from within the class variance multiplied with corresponding Eigen values (Belhumeur 1997). However, LDA uses supervised learning and emphasizes global structure because the denominator in Equation (6.6) is to be minimized whereas the numerator is to be maximized.

6.2.3 Locality Preserving Projection (LPP)

LPP tries to preserve the local structure or geometry of an image in the sampled data.

Let y_i be the one-dimensional mapping of x_i ($x \rightarrow w^T x$) and ($y \rightarrow w^T y$) that provides the best preservation of the local structure in the underlying distribution. It can be obtained by:

$$\min_w E(\ |w^T x - w^T y|^2 |\ \|x - y\| < \varepsilon) \quad \text{..... Eq (6.7)}$$

Let $z = x - y$. Then, Equation (6.7) can be written as:

$$\min_w E(\ |w^T z|^2 |\ \|z\| < \varepsilon) \quad \text{..... Eq (6.8)}$$

where w is a transformation matrix that can be treated as constant and defines a new indicator function $U_{ij,}$as follows:

$$U_{ij} = \begin{cases} 1 & \|x_i - x\|^2 < \varepsilon \\ 0 & otherwise \end{cases} \quad \text{..... Eq (6.9)}$$

Further, defining a diagonal matrix $D_{ii} = \sum_j U_{ij}$ and using estimation theory to solve Equation (6.8):

$E(z) = \frac{2}{d}(XDX^T - XUX^T)$, where E is the estimation of a large number and $z=x-y$

$E(z) = \frac{2}{d} XLX^T$, where $L= D-U$ is a Laplacian matrix and the i^{th} column of matrix X is x_i.

6.2.4 Orthogonal Laplacian Faces

Let $a \in \mathbb{R}^m$ be a projective map. The locality function can be defined as follows:

$$f(a) = \frac{a^T XLX^T a}{a^T XDX^T a} \quad \text{..... Eq (6.10)}$$

The orthogonal basis vectors $a_1, a_2, a_3, \ldots\ldots, a_k$ can be found by minimizing Equation (6.10) with the constraint $a_k^T a_{1=} a_k^T a_{2=} a_k^T a_{3=} \ldots\ldots = a_k^T a_{k-1=} 0$. Where

$$a_1 = \quad \arg_a^{min} \frac{a_1^T X L X^T a_1}{a_1^T X D X^T a_1}$$

$$a_2 = \quad \arg_a^{min} \frac{a_2^T X L X^T a_2}{a_2^T X D X^T a_2}$$

:

$$a_k = \quad \arg_a^{min} \frac{a_k^T X L X^T a_k}{a_k^T X D X^T a_k}$$

This minimization has been accomplished with a Lagrange multiplier and with the additional constraint $a_k^T X D X^T a_k = 1$to get

$$\lambda_k = \frac{a_k^T X L X^T a_k}{a_k^T X D X^T a_k}$$

6.3 FDC Faces

Histogram features can be combined with kernel learning methods to obtain excellent results in face recognition (Phillips 2000, Gao 2007). Local binary pattern (LBP) is one of the histogram feature extraction techniques in spatial domain which can capture the geometry of a face object and can provide efficient face recognition (Ahonen 2006). Gabor pattern combined with LBP has shown great results in face recognition using FERET database (Zhang 2005-b). Recently, kernel learning of histogram of local Gabor phase pattern (K-HLGPP) based on Daugman technique has been used in iris and palm print recognition showed high accuracy in face recognition as well (Daugman 1993, Zhang 2008).

Nearest neighbor classifier with chi-square/histogram intersection has also been used for face recognition (Barla 2003). Furthermore, Support Vector Machine (SVM) classifier uses Gabor wavelet (GW) to achieve positive results in recognition

(Belongie 2002). Another classifier, Bayesian is also used for face recognition (Martinez 2001, Moghaddam 1997, Duda 2001, Phillips 1998).

None of the techniques mentioned in the preceding paragraphs demonstrated higher accuracy while consuming lesser time than $O(n^2)$ especially with few training images. In this research, we develop a new classifier based on cumulative frequency distribution by using standard variance vector (SVV). SVV acts as a template kernel and is represented as a graph for GMT.

6.3.1 Classifier for FDC

Figure-6.1 shows cumulative frequency against gray levels. The figure is ploted using three images of the same class and three images from different classes. It is observed that plots belonging to the same class have almost identical profile. These plots can be divided into three distinct regions namely J_1, J_2 and J_3 as shown in Figure 6.1. These regions can be used to define three different threshold values corresponding to each region. This should not increase the search time because each region is computed independently.

Let $T = \{x_1, x_2, \ldots\ldots, x_m\}$ be a training set having m classes/columns. To mitigate the illumination effects, Mittal's relation for normalization of test and train images was defined in Section 3.2 is used. Define a transformation h on T which computes the gray level distribution as follows:

$h\left(x_{ij}\right) = n_{ij}$, where i $= 1,2, \ldots, m$ and $j = 0,1,2, \ldots,255$ …… Eq (6.11)

We then normalize the distribution:

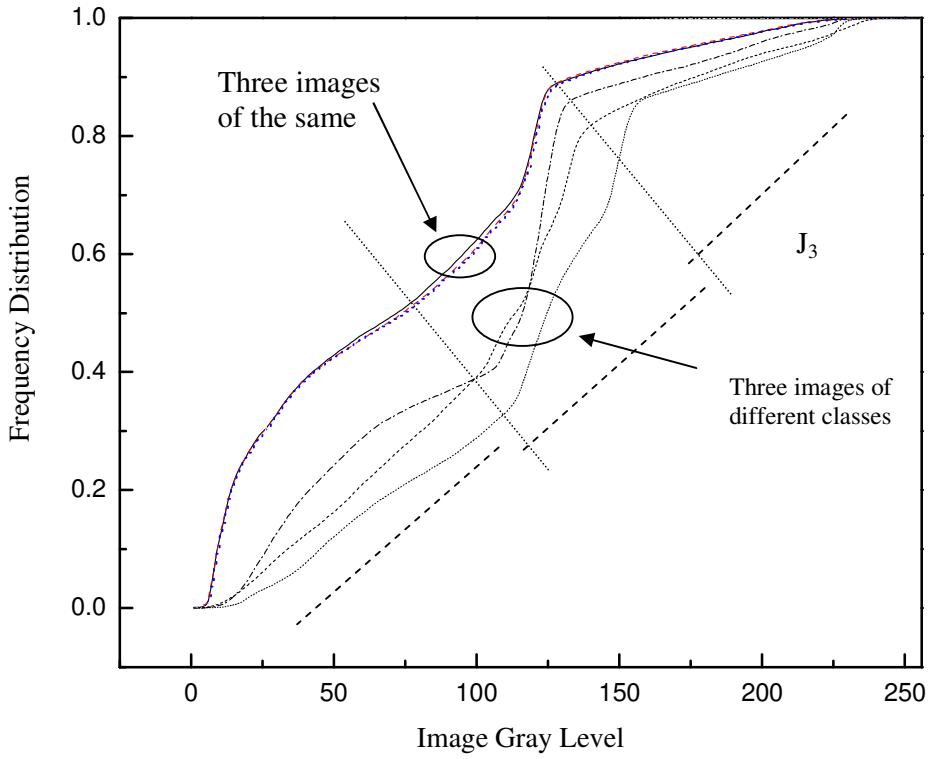


Figure-6.1 Shows distribution of gray levels for images of the same class compared with those images belong to different classes.

J_2

$$hn_{ij} = \frac{n_{ij}}{rc} \quad \text{..... Eq (6.12)}$$

where rc defines the resolution of an image. We accumulate the distribution as calculated in Equation (6.11) which provides a gradually increasing order of frequencies of gray level appearances. Whereas, each row of the normalized hn matrix represents a reference for the respective class.

Let ht be a transformed vector obtained by applying Equation (6.11) and (6.12), which is subtracted from each row of the hn matrix to populate a vector *M*.

$M_{iN} = hn_{ij} - ht_{Nj,}$ where N is the number of test images ..... Eq (6.13)

To maximize the variance between classes, each vector in Equation (6.13) is divided into three regions. Based on this concept the following objective function has been developed:

$$U_{ij} = \begin{cases} 1 & std(M_{iN_{j_1}}) < \varepsilon_1 and\ std(M_{iN_{j_2}}) < \varepsilon_2\ and\ std(M_{iN_{j_3}}) < \varepsilon_3) \\ 0 & \text{otherwise} \end{cases} \quad \text{..... Eq (6.14)}$$

$$\text{Where } \varepsilon_{1,}\ \varepsilon_2,\ \varepsilon_3 \in \{0,1\}$$

The lengths j_1, j_2, *and* j_3 as shown in Figure 6.1 have to be defined in such a way that the computed thresholds can be compared with $\varepsilon_{1,}\ \varepsilon_2$ and ε_3. It has been observed that $j1+j3 <= j2$ provides better results under the constraint $j1 + j2 + j3 = 255$.

6.4 Software Based System Performance

Many experiments have been performed to evaluate the performance of the proposed technique. For this purpose, the most popular holistic as well as feature-based techniques; namely PCA, LDA, LPP, and OLPP, are selected for comparison. Eigen face (PCA) and Fisherface (PCA+LDA) are holistic techniques, whereas,

Laplacianface (PCA + LPP) and O-Laplacianface (LPP + orthogonality) belong to the set of feature-based techniques.

Three popular face benchmark databases were used to validate the proposed technique. They are Yale. ORL, and PIE from the CMU databases (Yale 2002, Sim 2003, ORL 2009).

Yale database contains 165 images of 15 subjects and each image of a subject shows a different facial expression or situation involving smile, sadness, eyeglasses or flash, as shown in Figure-6.2.

Train2, Train3, Train4 and Train5 represent the training sets from Yale database and contain a randomly selected set of 2, 3, 4 and 5 images per subject, respectively. All the remaining images are used for testing in the experiments. The false acceptance rate (FAR) and false rejection rate (FRR) were defined in Section 3.5 and they are used to measure the minimum error rate (MER). Furthermore, MER was discussed in Section 3.7 and is treated as error rate (ER) in this chapter. To remove stochastic errors, an average value of the error is reported. The number of dimensions is equal to the number of feature vectors used in an experiment.

Figure-6.3 shows ER profiles of PCA, LDA, LPP, OLPP and the proposed technique for Yale database. The accuracy of PCA, LDA, LPP and OLPP increases by increasing the dimension of a given training set. The figure also shows that the accuracy for the same dimensionality improves, when the system is trained with large training sets. For example, PCA shows ~55% and ~45% ER at 25 dim in Figure-6.3 (a) and 6.3(d), respectively.

Figure-6.2 Sample images of a subject from Yale face database.

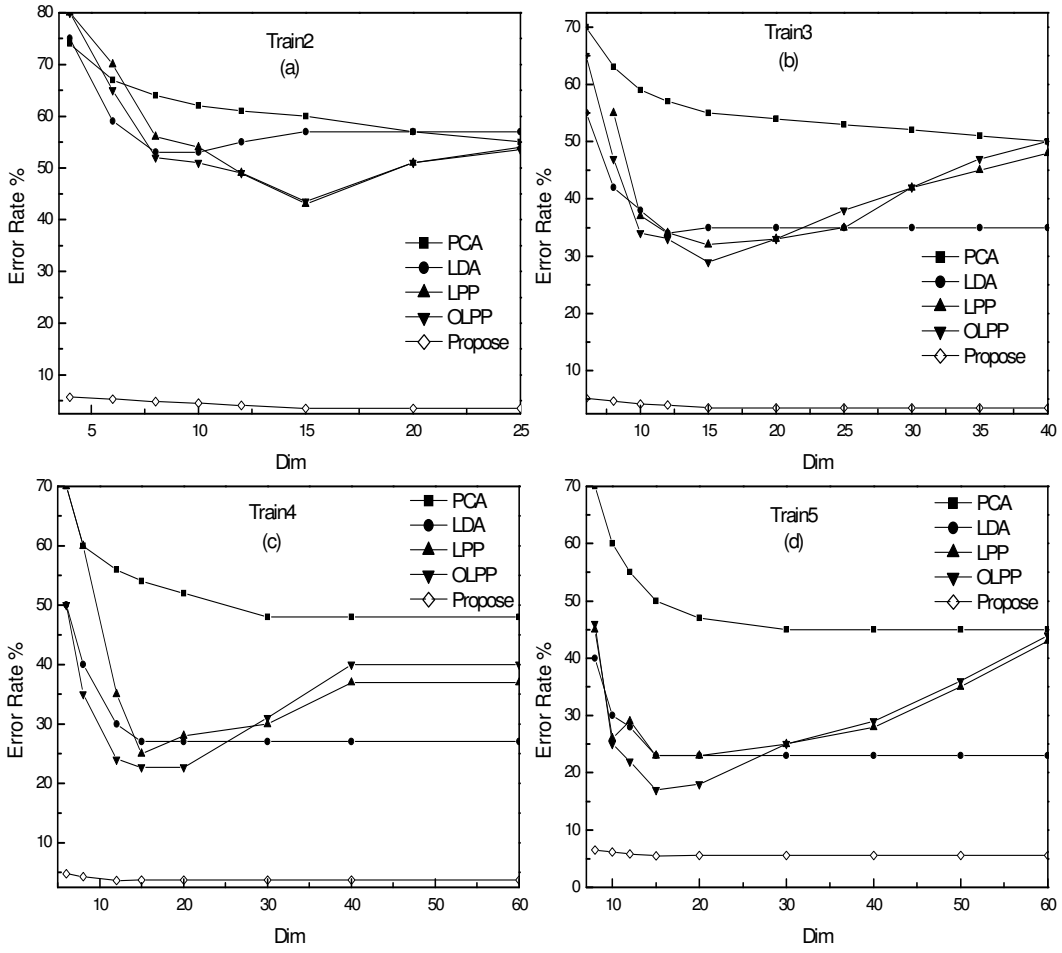


Figure-6.3 Error rates of PCA, LDA, LPP, OLPP and the proposed technique for Yale database.

Table-6.1 MER comparison of PCA, LDA, LPP, OLPP and the proposed technique for Yale database.

Technique	Training Set			
	Train2	Train3	Train4	Train5
PCA	56.5%(25)	51.1%(44)	47.8%(58)	**45.2%(71)**
LDA	54.3%(9)	35.5%(13)	27.3%(14)	**22.5%(14)**
LPP	43.5%(14)	31.5%(14)	25.4%(14)	**21.7%(14)**
OLPP	44.3%(14)	29.9%(14)	22.7%(15)	**17.9%(14)**
Proposed	**3.5%(15)**	3.5%(15)	3.61(12)	5.53%(15)

The ER curve for LDA is different than that of PCA, LPP and OLPP but is quite similar to that of proposed technique. The LDA accuracy initially improves with an increase in the dimensions, but after a certain dimensionality it becomes constant.

This trend can also be seen in the proposed technique however the decline rate and the initial values of ER are different. On the other hand, LPP and OLPP show convex profiles and their curvature becomes deeper (more accurate) for larger training sets as shown in Figure-6.3 (c & d). Figure-6.3 clearly demonstrates that the proposed technique has the lowest ER for all training sets and is almost independent from dimensionality.

Table 6.1 shows MER values calculated from Figure-6.3 for PCA, LDA, LPP, OLPP. It clearly demonstrates that the proposed technique provides significantly better accuracy relative to its counterparts. ORL face database contains 400 images of 40 subjects. These images have variations in facial expression, like open or closed eyes, with or without smile, and extra facial features like glasses etc. Furthermore, the images may show face tilting and/or rotation, as shown in Figure-6.4. The training sets are grouped as Train2, Train3, Train4 and Train5 that comprise 2, 3, 4 and 5 randomly selected images per subject respectively. Experiments for each training set were performed while the remaining images of the database were used for testing.

ER values for different training sets are shown in Figure 6.5, whereas, Table 6.2 represents MER for the same database. This once again demonstrated that the proposed technique is far better in accuracy than its counterparts. The observed

MER for ORL database is better than Yale and it is almost independent of Training data set.

PIE database contains 11,560 images of 68 subjects (Sim 2003). These images were captured using 13 cameras operating simultaneously with 21 flashes. They have low resolution and contain various facial positions and expressions effects as shown in Figure-6.6.

Four training sets were randomly selected namely: Train5, Train10, Train20 and Train30 containing 5, 10, 20, and 30 images respectively. The remaining images were used for testing. Thus, our experiments involved exhaustive testing by employing a huge database and results are shown in figure-6.7.

The same FAR and FRR error strategies were used to calculate ER. For Train2, PCA starts with a 100% error, then improves gradually and finally shows an MER of 69.9% at 338 dim. Whereas, LDA rapidly improves its ER from 70% to 31.5% at 67 dim. LPP and OLPP start with similar ER (about 40%) but OLPP improves rapidly and attains 21.4% ER, whereas, LPP reaches to 30% ER at 67 dim. It is observed that ER values improve by increasing images in the training sets.

The ER 21.4% is the MER of OLLP with 108 dimensions and represents the minimum among PCA, LDA and LPP. On the other hand, the proposed technique starts with a 3.5% ER and reaches a 3.12% MER with 136 dimensions for PIE database as shown in Figure-6.7 and Table-6.3.

The observed MER values for different training sets are shown in Table 6.3. The data of the table show that the proposed technique yields 3.12%, 3.12%, 8.74% and 8.58% MER for dimensions 136, 272, 544 and 696, respectively.

Figure-6.4 Sample face images of a subject from ORL

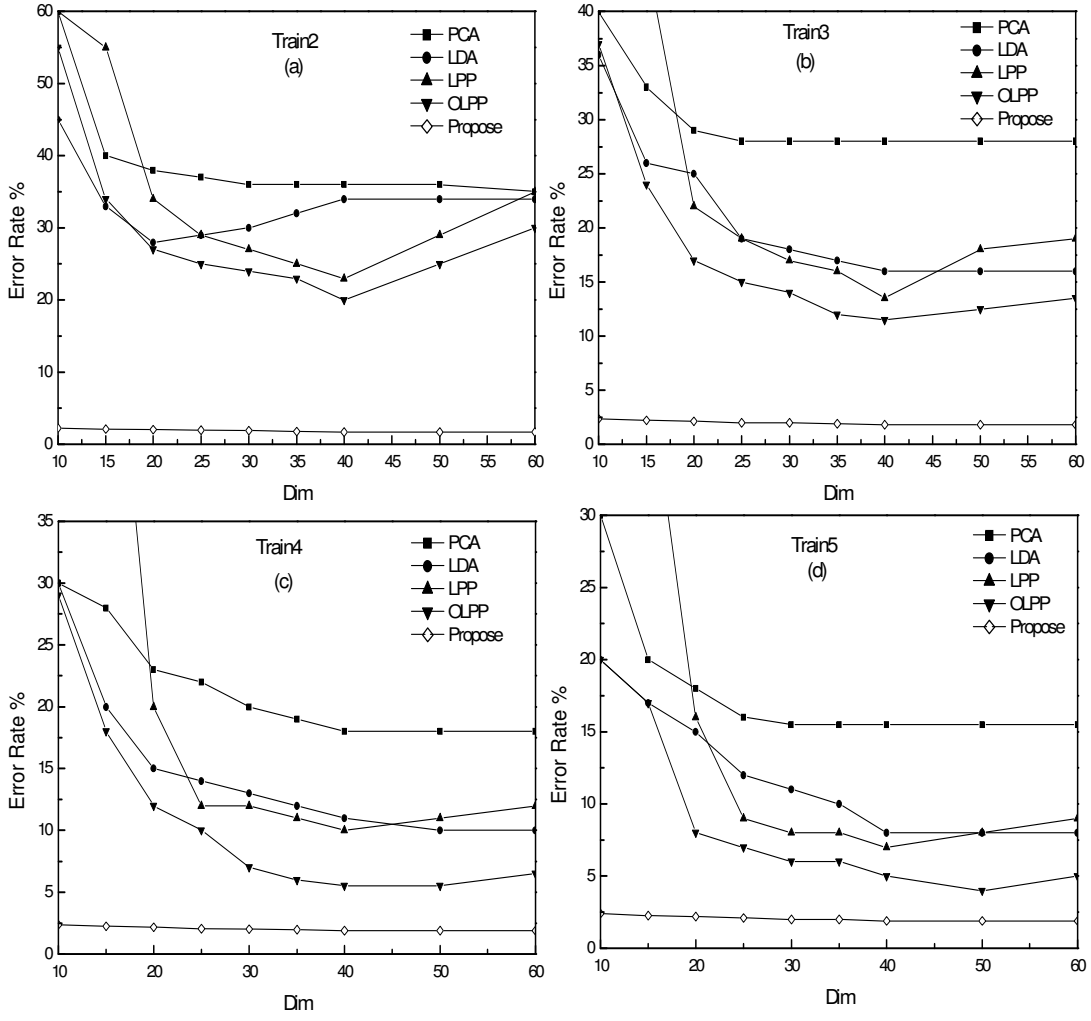


Figure-6.5 Error rate of PCA, LDA, LPP, OLPP and the proposed technique for ORL database.

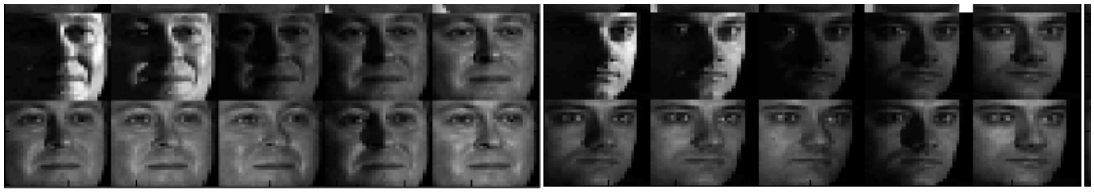


Figure-6.6 shows ten images per subject for two subjects from PIE database.

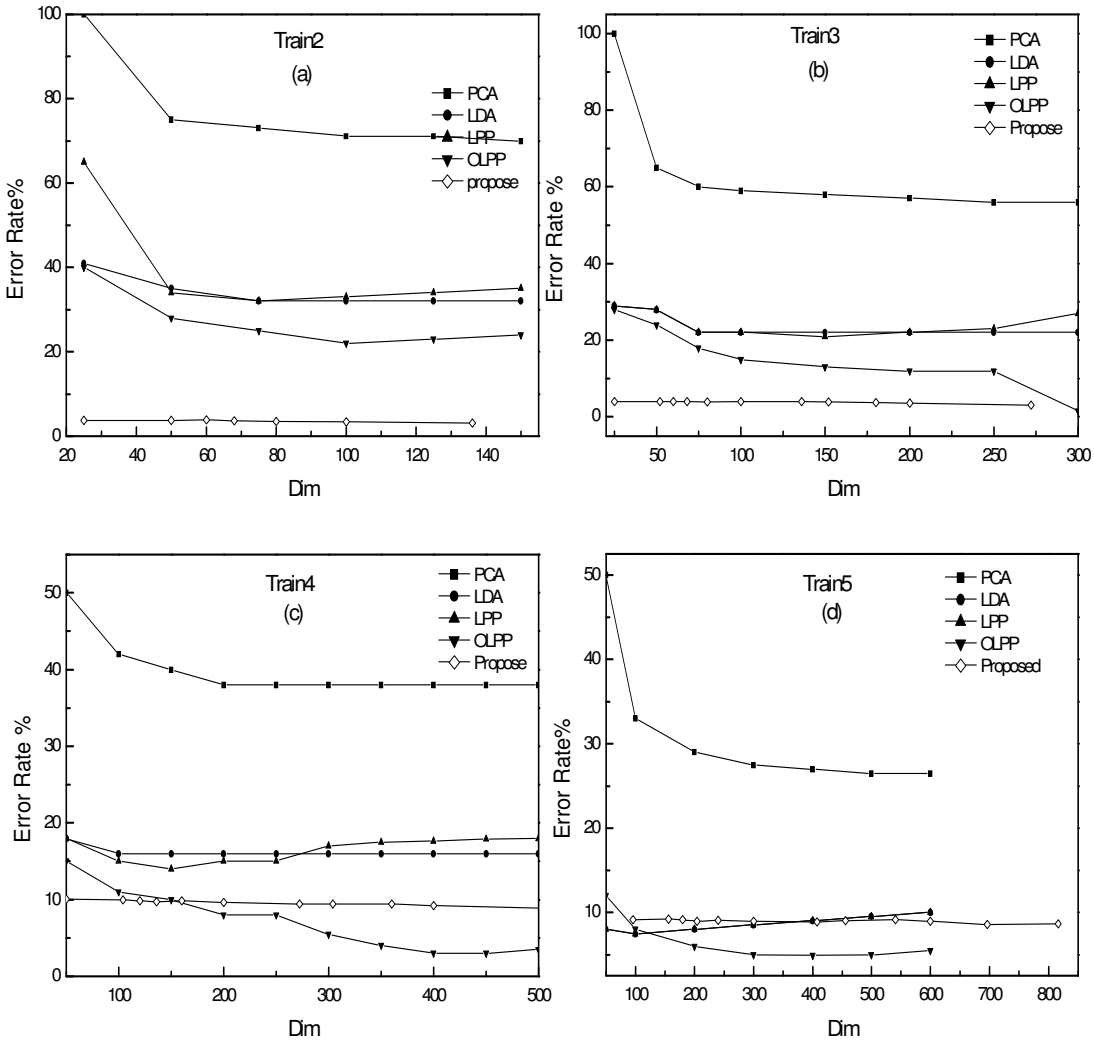


Figure-6.7 Error rates of PCA, LDA, LPP, OLPP and the proposed technique for PIE database.

Table-6.2 MER of PCA, LDA, LPP, OLPP and the proposed technique for ORL database.

Technique	Training Set			
	Train2	Train3	Train4	Train5
PCA	33.7%(78)	24.6%(119)	18%(159)	**14.1%(199)**
LDA	28.9%(22)	15.8%(39)	10.5%(39)	**7.75%(39)**
LPP	23.9%(39)	13.4%(39)	9.58%(39)	**6.85%(40)**
OLPP	20.4%(40)	11.4%(39)	5.92%(48)	**3.65%(59)**
Proposed	**1.7%(40)**	1.79%(40)	1.9%(40)	1.9%(40)

Table-6.3 MER of the PCA, LDA, LPP, OLPP and the proposed technique for PIE database.

Technique	Training Set			
	Train2	Train3	Train4	Train5
PCA	69.9%(338)	55.7%(654)	38.1%(889)	**27.9%(990)**
LDA	31.5%(67)	22.4%(67)	15.4%(67)	**7.77%(67)**
LPP	30.8%(67)	21.1%(134)	14.1%(146)	**7.13%(131)**
OLPP	21.4%(108)	11.4%(265)	6.51%(493)	**4.83%(423)**
Proposed	**3.12%(136)**	3.12%(272)	8.74%(544)	8.58%(696)

The table shows that the proposed technique's performance deteriorated for Train4 and Train5 at high dimensions.

It can be observed, in general, that the ER for PIE database is higher than that of Yale and ORL databases. Because, PIE contains 170 different low-resolution images per person. Using Yale and ORL, the proposed technique outperforms regardless of the training set and dimensions. For PIE, our technique provides better results with Train2 and Train3; with Train4 and Train5, it remained better for dimensions less than 150 and 100 respectively, as shown in Figure-6.7 (c & d).

There is a saturation point in relation to the dimensions after which our technique generates a larger ER. It is observed that this saturation point is reached earlier for a bigger training set. Thus, the proposed technique provides better results for smaller training sets with dimensions equal to the number of classes because the decision space created by our technique is based on cumulative frequency vectors. Whereas the Eigen space usually used by other techniques is based on basis vectors.

The Eigen space provides better results when including more basis vectors while the proposed technique requires optimally number of vectors equal to the number of classes (as for LDA). However, sometimes a minor improvement can be seen in the proposed technique by increasing the dimensions, like in Figure-6.7 (a) where its MER is 3.62% for 68 dimensions whereas its ER is 3.12% for 136 dimensions. Figure-6.3 and Figure-6.5 show the MER values of the proposed technique with training sets which are smaller in the cases of ORL and Yale as compared to PIE. The proposed technique performance is superior for small training set, ideally one, and dimensions equal to the number of classe to facilitate needs of a real-time System.

6.5 FPGA Based FDC System Architecture

Our FDC-based three-layer architecture is shown in Figure-6.8. The feature extraction is a computationally intensive part but it is accomplished once, therefore, handled offline during pre-processing. This process produces PVs which are stored in the host machine. One PV is transferred at a time to the online layer either by PCI or by any other bus interface.

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 when transferred to an FPGA. Such a bandwidth may not be available due to system constraints. For this reason, our approach is to convert the image matrix into one-dimensional IPV which is subsequently transferred to an FPGA even using a low bandwidth data bay. The data is received by an FPGA buffer which directs this to the on-chip BRAM memory for FDC computation by using Equations (6.11) and (6.12).

One PV from the pre-stored collection of PVs in the host RAM is transferred and compared with IPV. The classifier as per Equations (6.13) to (6.14) determines statistically 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. The efficiency of the architecture was further improved (Figure-6.8) by introducing an adaptive classification technique for the pre-stored (training set based) PVs.

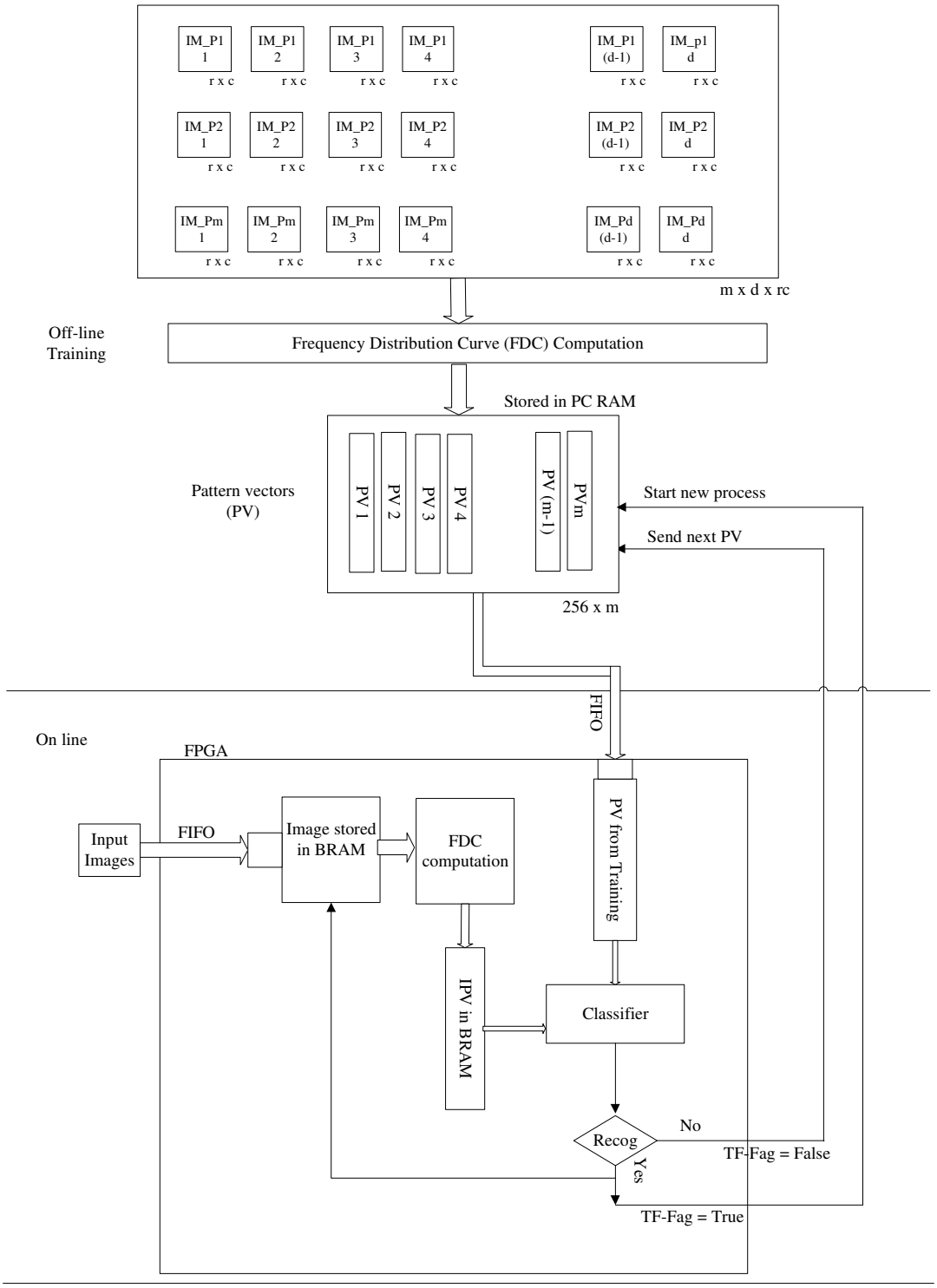


Figure-6.8 FDC-based face recognition architecture.

The enhanced architecture relies on a process that eliminates the need to test all the pre-stored PVs against the online produced IPV. The new architecture is shown in Figure-6.9. In this architecture, the Euclidian normalization length of a vector is used to calculate tags for pre-stored 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

$x = \left[\sum_{i=1}^{256}\left(abs(x_i^2)\right)\right]^{1/2}$ *calculate tags for the stored PVs.*

Step (2): The computed tags for PVs are stored in a vector $V_m = x_j$,

$where\ 1 \leq j \leq m$

Step (3): Apply step (1) for the test image and get a 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 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 a PV having 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 is attained.

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

6.5.1 Adaptive Controller for FDC

In Figure-6.9, the host layer has a tag vector, V_m 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 buffer to get a new test image. Furthermore, for a *true* signal, a new ID_IPV is sent to the host layer.

Simulation results in histogram form of the new algorithm using ORL database are shown in Figure-6.10. The column length indicates the frequency of successful attempts, whereas on the horizontal axis, corresponding PVs are shown. The first column in Figure-6.10 shows that ~190 test images will get a successful PV from the PV_controller within fifth iteration. And the average number of iteration required to get a successful PV to 7.7 from 20. Controller has improved the architecture originally conceived in Figure 6.8. Adaptive controller works in host layer, online layer performance is independent of the adaptive controller computational and storage cost.

6.6 FDC Parallel Processing Architecture

In Figure-6.11 a parallel processing architecture is formulated using ID_PV and V_m components. In this architecture, PV_controller is replaced with a new layer to attain PV classification. This layer has three major components, namely, Trained_PV; P_PV_controller and PV_Cache. Trained_PV represents the collection of PVs and P_PV_Controller has two extra components compared to the architecture shown in Figure-6.9. These components are circular FIFO and a k-dimensional array containing the sorted tags of PVs in the corresponding parallel processing modules.

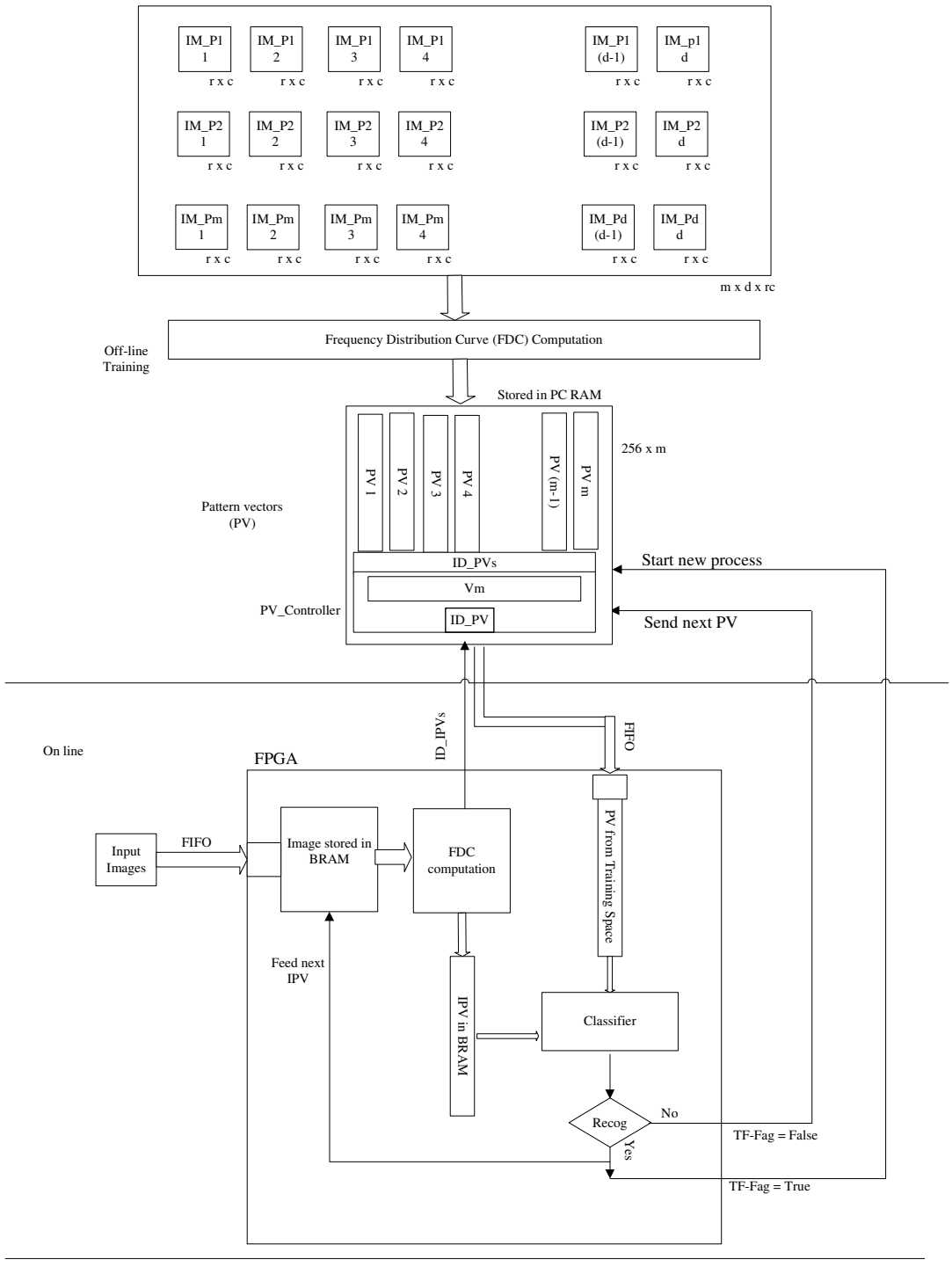


Figure-6.9 FDC-based face recognition architecture using adaptive controller.

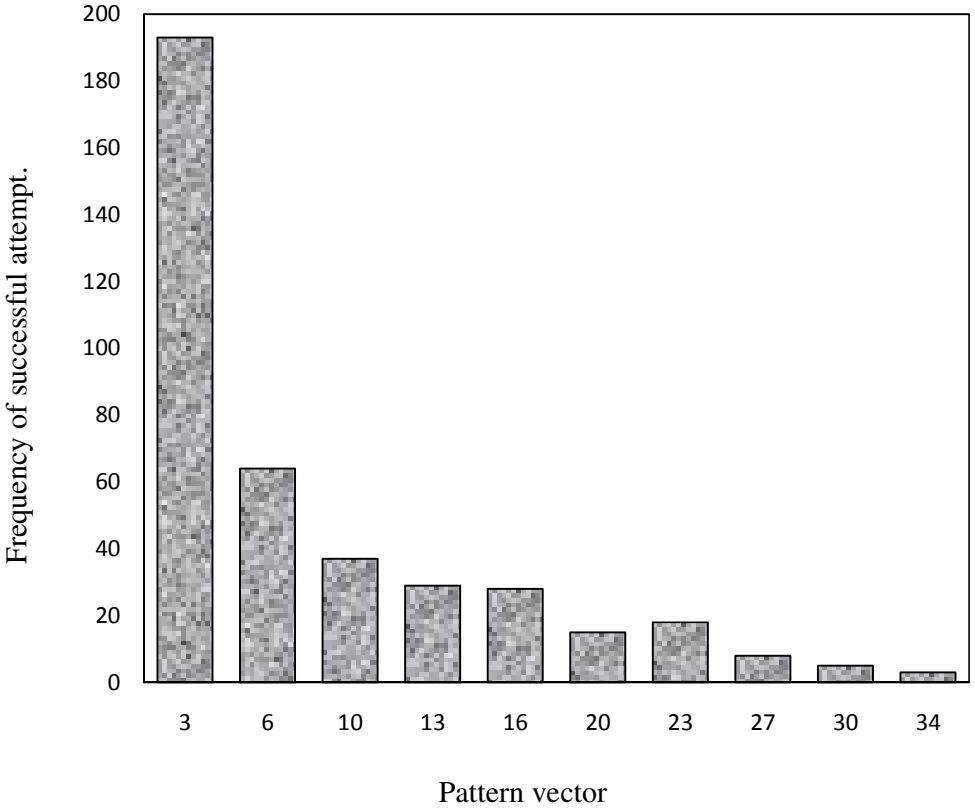


Figure-6.10 Simulation results showing the high probability of success virus pattern vector for ORL database.

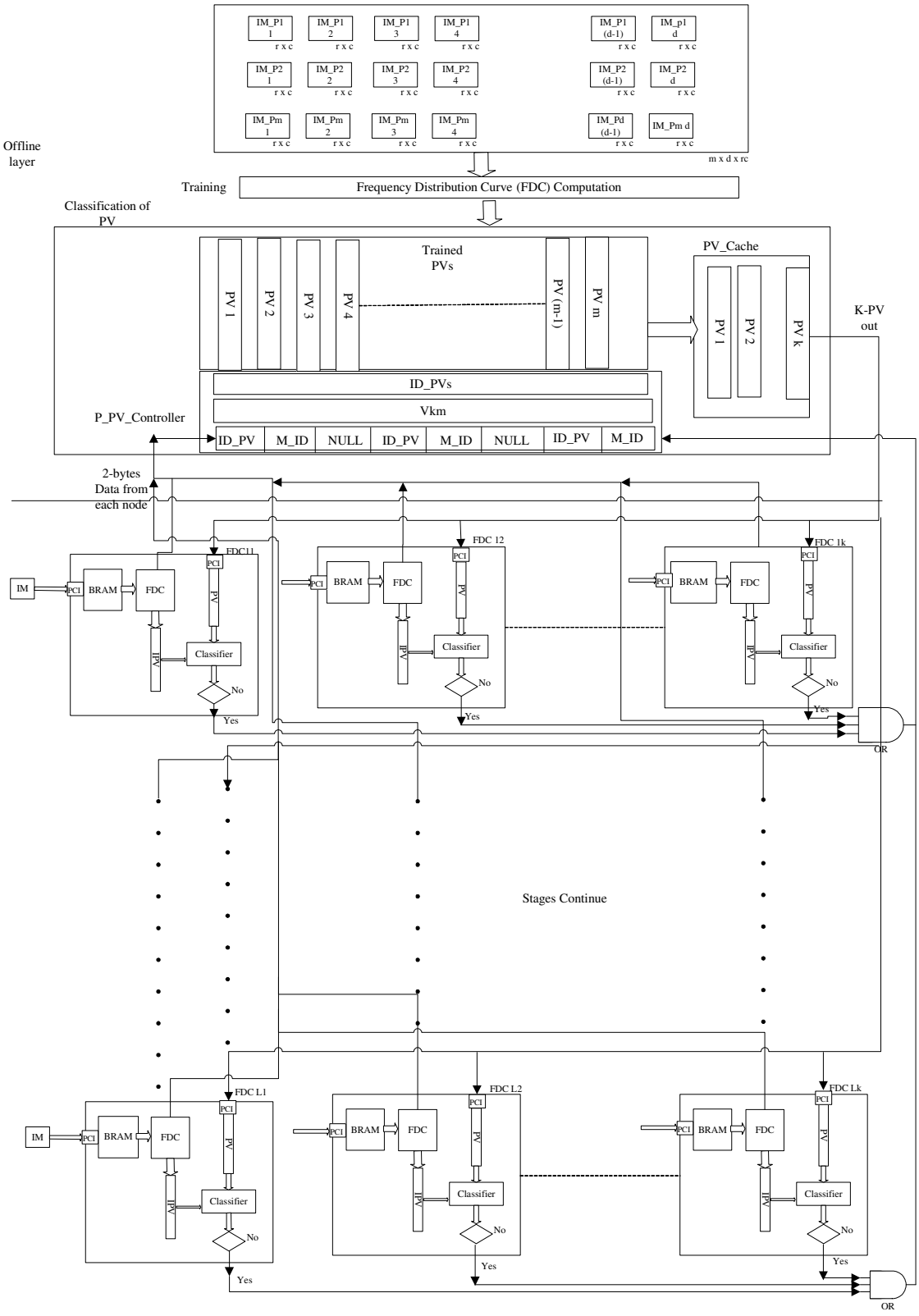


Figure-6.11FDC parallel processing face recognition architecture.

Circular FIFO is introduced in PV_Controller to handle the sequence of identification tags for incoming test images. It is ensured that k modules should get most suitable trained PVs from the controller layer which are stored in advance in PV_Cache area.

The diamond block provides input to the OR gates for further processing after a successful match/recognition. This block generates a positive decision if the test image belongs to a person from the training database; otherwise, its feedback signal to the host machine requests the next appropriate PV. This block plays an important role in the parallel processing architecture using the PV_Cache area, as shown in Figure-6.11.

A *true* answer for the Boolean expression shown in Equation (6.14) 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-6.11 depend upon the target FPGA resources, and the vertical elements determine the number of FPGAs on the target board. Therefore, the proposed parallel processing architecture is a modular and a robust reconfigurable system. The speed of the obtained architecture depends on the number of stages involved and by assuming a 10% overhead for new stages; the simulated performance of the proposed architecture is shown in Figure- 6.10.

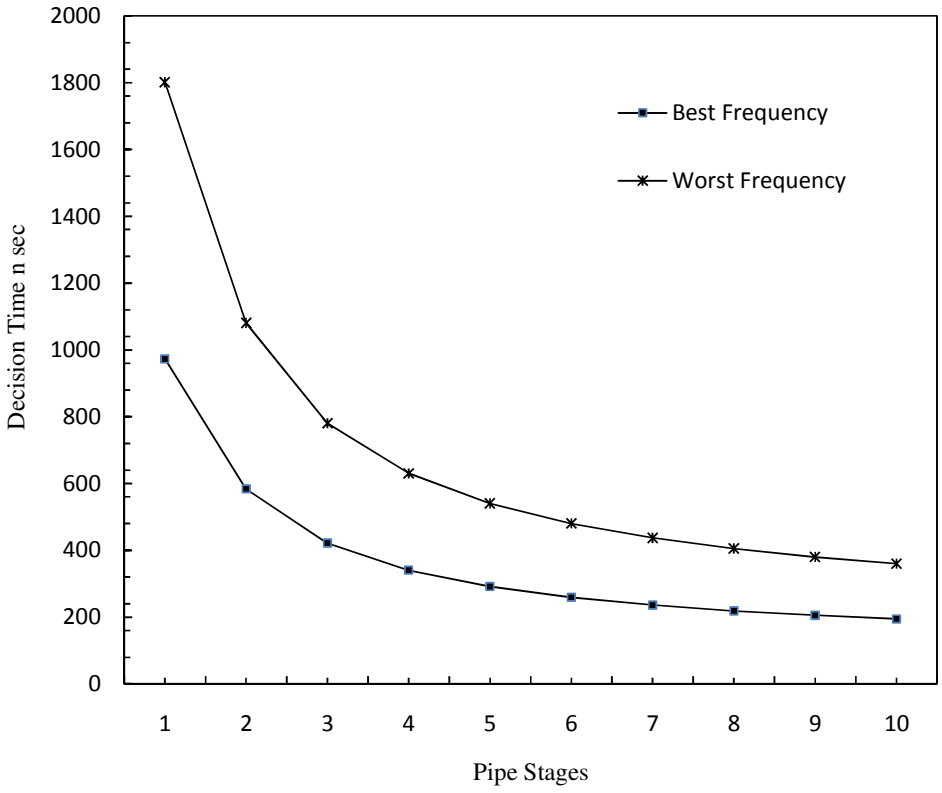


Figure-6.12 Pipelining performance versus number of stages.

6.7 System Performance

Table-6.4 shows the resource utilization data of our architecture shown in Figure- 6.8 when implementing on a Xilinx Virtex5 FPGA device. The Xilinx 11.1 suite was used which includes the AccelDSP tool (AccelDSP 2008) with MATLAB files for floating-point verification. Pipelining in multipliers, adders and subtractors implementation further reduces the time but the resource consumption reaches 36% of the FPGA real estate.

The Xilinx development tool provides two frequencies to design a system, one being the requested frequency and other the maximum frequency based on the circuit's critical path after RTL implementation. The execution time for both the frequencies is shown in Table-6.5 using ORL database. The parallel processing architecture (Figure-6.11) shows an improvement up to 83 % compared to the architecture in Figure-6.8.

Table-6.5 indicates a significant improvement in decision time of the proposed parallel processing architecture. Matching an input image with stored PVs takes ~2527 (126 x 20) ns for non-parallel processing architecture because it needs on the average twenty iteration for matching. The number of iteration reduces to ~7 after introducing a controller in the architecture as shown in Figure- 6.10. Thus, architecture with controller reduces the decision time 2527 ns to 973 ns for matching. Three-stage parallel processing architecture with controller has achieved decision time ~421 ns. Figure- 6.12 shows that by increasing parallel processing stages the decision time reduces, but after nine stages it almost becomes constant.

Table-6.4 Resource utilization of proposed architecture using ORL database on Virtex5 (XC5VSX50T).

Resources	Quantities
Slices	36%
BRAM	1%
Multipliers	15
Adders	53
Subtractors	47
DSP units	19%

Table-6.5 Performance of proposed architectures using ORL database on Virtex5 (XC5VSX50T).

Option	Best	Worst
Frequency in MHz	188	100
Data feeding time (ns)	118.72	224
Hardware time (ns)	7.66	9.96
Data out time (ns)	0.0053	0.01
Time require for matching with a PV (ns)	126.38	233.97
Total testing time without controller (ns)	2527.70	4679.4
Total testing time using adaptive controller (ns)	973.16	1801.57
Total testing time using parallel processing architecture (ns)	421.71	780.68
Improvement with parallel processing architecture % (100-(PA/WC)*100)	83.316	83.316

PA= Parallel processing architecture
WC= Without controller

6.8 Summary

Non-intrusive biometrics is a complex task, especially for face recognition because the input images are usually not well classified and managed. But it is probably the only automated way to detect a suspect. High accuracy with minimum decision time is a challenging job for real-time face recognition systems. Principal component analysis (PCA) is a classical approach amongst the dimension-reduced feature extraction methods. Its accuracy is poor because it treats faces as global entities. The local preserving projection (LPP) and orthogonal local preserving projection (OLPP) methods treat faces as combinations of features, in addition to dealing with global structures. LPP and OLPP do not provide a very high accuracy (in the range of ~98%) even when using big training sets because the reduced space may downplay some critical information.

We have proposed a face recognition technique which extracts global structural information and local details using gray level frequency distribution curve (FDC). The FDC provides better results for smaller training sets with dimensions equal to the number of classes. By using Yale, ORL and PIE databases, it has been demonstrated that the proposed technique is better both in terms of accuracy and time. The proposed technique is then implemented on FPGA using three layer architecture decision time 2.52 µs which is faster than a PC based recognition system. The architecture was further improved by designing an adaptive controller which gave 40% improvement relative to non-controller based architecture. Furthermore, the parallel processing architecture exploits the parallelism capabilities of FPGA devices, thus providing decision in less than 200 ns and making the system a good candidate for real-time face recognition.

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 Conclusion

Biometric plays an important role in enhancing efficiency and reliability of an authentication system. Amongst the most common biometrics, fingerprints and iris recognitions are intrusive in nature, while voice and face recognitions are non-intrusive types of biometrics. Significant efforts have been made by the researchers to improve accuracy and efficiency of a face recognition system to meet the stringent needs of defense, security and high-tech commercial applications. Face recognition systems commonly employ algorithms which provide reduced computational space and meaningful data in the form of Eigen values.

In PCA Eigen values of a co-variance matrix are evaluated, and corresponding Eigen vectors are used to create an *n*-dimensional space, where *n* is the rank of a co-variance matrix. This *n*-dimensional space is less computational intensive than the original image space. Another important role of PCA is to maximize the variance in reduced feature space and thus it provides image classification by using few Eigen vectors.

For real-time face recognition system in a dynamic environment where the face database may change and learning is a repetitive process, fast calculation of Eigen values and vectors is a primary requirement. Fast PCA has been proposed for efficient generation of Eigen values which improves the computational efficiency to O (n^2) compared to normal decomposition method which gives the solution in O (n^3) time. In fast PCA, however, non-convergence state may be observed especially, in those cases, where high resolution images are involved. This could be associated with GSO which

is used in fast PCA to evaluate Eigen value. To overcome this problem, fast PCA has been modified by incorporating MGSO to generate Eigen values for images including those at high resolutions.

The accuracy of a PCA based face recognition system depends upon the performance of its classifier which improves by increasing the number of Eigen vectors involved in the decision making. However, increased Eigen vectors make the system computational intensive. Thus, an improved performance of a face recognition system, within requisite decision time, is dependent upon the optimum selection of Eigen values in a decision making process.

Adaptive classifier has been developed for the modified fast PCA. This has the ability to tune itself by varying threshold value during learning process and select optimum number of LEVs. Further, a system variable searcher module has been designed using two dimensional data structure to achieve the best combination of system variables for its enhanced reliability and accuracy. The validity of the proposed system has been checked by varying Eigen vectors and training sets. The modified fast PCA demonstrated 8% improvement in accuracy and it was three times faster compared to decomposition based PCA.

For Eigen values evaluation of a symmetric matrix, decomposition is believed to be a superior method compared to Jacobi transformation. There are many decomposition methods, out of those, Householder (HH) is considered an efficient one. This is primarily due to the fact that Householder evaluates square root by using non-restoring algorithm instead of trigonometric functions. CORDIC-based Jacobi

algorithm (CJA) is one of the best reported FPGA implementation which provides Eigen values by involving trigonometric functions for square root evaluation.

To achieve cost and time effective solution, HH was implemented by adopting a co-design methodology. In this connection, a co-design pipelined architecture has been developed by employing fixed-point non-restoring algorithm instead of trigonometric functions. The proposed architecture demonstrated an improvement up to 30% in time and 10^{-7} decimal places in accuracy compared to CJA.

Modified fast PCA uses MGSO which applies normalization of vectors in its iterative orthogonal process. Three architectures have been developed to achieve efficient normalization on an FPGA. The validity of these architectures is then demonstrated with ORL, PIE, FERET and CAS-PEAL databases. An analysis has been presented to show the suitability of an architecture for a given application. In all these architectures, a maximum observed error was 10^{-3} compared to their software-driven counterpart, when implemented in floating-point.

High accuracy with minimum decision time is a challenging task for a real-time face recognition system. PCA is a classical approach amongst the dimension-reduced feature extraction methods towards efficient recognition. However, PCA offers poor accuracy because it treats faces as global entities. The local preserving projection (LPP) and orthogonal local preserving projection (OLPP) treat faces as combinations of features, in addition to dealing with global structures. A frequency distribution curve (FDC) technique has been developed which transforms linearly a high dimensional image space to 2-dimensional face space. FDC has the ability to preserve global along with major local face details and can still make a decision in

substantially reduced time, *i.e., O* (n^2). This is associated with the fact that FDC avoids matrix decomposition and other high computational matrix operations and extracts features using pattern vector (PV). One or two training images per subject are sufficient for FDC to compute PVs. A software implementation of FDC showed an improvement up to 14%, 1.9% and 1.7% for the Yale, ORL and PIE face databases respectively compared to PCA, LPP and OLPP.

FDC is formulated with a bias towards its efficient hardware realization. An FPGA-based architecture has been developed by designing an adaptive PV controller. This controller has the ability to detect high probability PV for immediate matching instead of sequential thorough checking. The architecture performance was further improved by incorporating parallelism which demonstrated an improvement up to 80% compared to sequential architecture. The system thus developed requires less than 200 ns to make a decision.

7.2 Future work

The proposed FDC matching technique divides the curve into three segments by treating it as Gaussian distribution. The decision about the length of these segments and their associated threshold values is not trivial. It has been observed that the threshold value varies considerably by changing image characteristics. On the other hand, a minor variation in the threshold value causes serious adverse effects on the recognition accuracy. It is an established fact that the threshold value is dependent upon the image characteristics like its resolution, illumination and posture etc.

It is suggested that an estimation technique like maximum likelihood, Cramer-Roa lower bound (CRLB) or Bayesian estimator (Kay 1993) may be applied on the proposed FDC technique to adjust the threshold as per image characteristics.

Further, the storage space for PVs in FDC technique grows linearly by increasing subjects in the training set which may lead to an inefficient system. To overcome this, an adaptive PVs controller (APVC) has been designed in a reconfigurable hardware. APVC uses vector normalization of PVs to compute the probability associated with these vectors. The data thus acquired are used for the identification of a high probability PV which allows selective rather sequential matching. APVC has shown good results for small to medium data sets. However, its performance deteriorates with enhanced input data set. Since, Hidden Markov Model (HMM) is one of the best stochastic techniques to provide the next sequence of event based on the previous knowledge (Li 2000). It is, therefore, suggested that HMM may be used for the selection of high probability PVs.

Fixed-point implementation of HH on FPGA provides a time-efficient solution for the evaluation of Eigen values. However, its accuracy and efficiency depends upon the selection of *q.n* format that varies with image characteristics. Therefore, an analysis for the trade-off between accuracy and efficiency could be a beneficial study for various image characteristics. This could provide an optimum combination between accuracy and efficiency for HH implementation. Furthermore, it is suggested that an adaptive controller (AC) might be designed which could provide a better adjustment for *q.n* format for changed image characteristics. It is expected that a system incorporating such an AC should further enhance the evaluation of Eigen values.

BIBLIOGRAPHY

(AccelDSP 2008) Accel DSP Synthesis Tool User Guide, 2008.

(Ahonen 2006) T. Ahonen, A. Hadid and M. pietikainen, "Face Description with local Binary Pattern: Application to Face Recognition", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 28, no. 9, pp. 1041-1050, 2003.

(Altera 2001) Altera Application note 74, "Evaluating Power for Altera Devices", 2001.

(Anton 2005) Anton, H., C. Rorres, "Elementary Linear Algebra, ninth edition", *Oxford University Press*, pp. 138, 2005.

(APU 2007) APU Floating-Point Unit v3, Xilinx, January 26, 2007. http://china.xilinx.com/support/documentation/ip_documentation/apu_fpu.pdf.

(ARM 2009) ARM code square root routines, www.finesse.demon.co.uk/steven/sqrt.html**,** Jan 2009.

(Barla 2003) A. Barla, F. Odone, and A. Verri, "Histogram Intersection Kernel for Image Classification," *Proceedings of IEEE international conference on image processing (ICIP '03)*, vol. 3, pp. 513–516, Spain, 2003.

(Barlett 1998) M. S. Barlett, H. M. Lades and T. J. Sejnowski, "Independent Component Representation for Face Recognition", Proc. of SPIE, pp. 528-539, 1998.

(Belhumeur 1997) P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711-720, 1997.

(Belkin 2001) M. Belkin and P. Niyogi, "Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering," Proc. Conf. Advances in Neural Information Processing System 15, 2001.

(Bell 1997) A.J. Bell and T.J. Sejnowski, "The Independent Components of Natural Scenes Are Edge Filters," Vision Research, vol. 37, no. 23, pp. 3327-3338, 1997.

(Belongie 2002) S. Belongie, C. Fowlkes, F. N. Chung, and J. Malik, "Spectral partitioning with indefinite kernels using the Nystorm extensions," in *Proceedings of the 7th European Conference on Computer Vision (ECCV '02)*, pp. 531–542, Copenhagen, Denmark, May 2002.

(Benkrid 2010) K. Benkrid, D. Crookes, A. Bouridane, P. Con, and K. Alotaibi, "A High Level Software Environment for FPGA Based Image Processing", *Queen's University of Belfast UK online document.*

(Biederman 1997) I. Biederman and P. Kalocsai, "Neurocomputational Bases of Object and Face Recognition", *Philosophical Transactions of the Royal Society: Biological Sciences,* vol. 352, pp.1203–19, 1997.

(Bjorck 1967) A. Bjorck, "Solving Linear Least Squares Problems by Gram-Schmidt Orthogonalization", *BIT Numerical Mathematics*, vol. 7, no. 1, 1967.

(Bjorck 1996) A. Bjorck, "Numerical Method for Least Squares Problems", *SIAM*, 1996.

(Blanz 1999) V. Blanz and T. Vetter, "A Morphable Model for the Synthesis of 3D Faces", *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pp. 187–194, 1999.

(Bravo 2006) I. Bravo, P. Jimenez, M. Mazo, J. L. Lazaro, and A. Gardel, "Implementation in FPGAS of Jacobi Method to Solve the Eigenvalue and Eigenvector Problem", *Proceeding of International Conference on Field Programmable Logic and Applications FPL*, p.p. 1-4, 2006.

(Brown 1996) S. Brown, and J. Rose, "Architecture of FPGAs and CPLDs: A Tutorial," *IEEE Computer Society Press Los Alamitos*, CA, USA, vol 13 , issue 2 Jun., 1996.

(Bruce 1988) V. Bruce, Recognizing Faces, *Lawrence Erlbaum Associates*, London, U.K, 1988.

(Buhmann 1990) J. Buhmann, M. Lades and C. V. D. Malsurg, "Size and Distortion Invariant Object Recognition by Hierarchical Graph Matching", *Proceedings, International Joint Conference on Neural Networks*, pp. 411–416, 1990.

(Burden 1997) R.L. Burden and J. D. Faires, seventh ed., "Numerical Analysis" *Thomson*, 1997.

(Cai 2006) D. Cai, X. He, J. Han and H. J. Zhang, "Orthogonal Laplacianfaces for Face Recognition", *IEEE Transaction on Image Processing,* vol.15, no.11, 2006.

(Cardoso 1998) J. F. Cardoso, "Blind Signal Separation, Statistical Principles", *IEEE Proceeding*, vol. 9, no. 10, pp. 2009-2025, 1998.

(Chang 2003) Y. Chang, C. Hu and M. Turk, "Manifold of facial Expression", IEEE Proceeding International Workshop Analysis and Modeling of Faces and Gestures, 2003.

(Chellappa 1995) R. Chellappa, C.L. Wilson and S. Sirohey, "Human and Machine Recognition of Faces: A Survey", *IEEE Proceeding* vol.83, no. 5, pp. 705-740, 1995.

(Chellapa 1997) K. Elemad and R. Chellappa, "Discriminant Analysis for Recognition of Human Faces Images", Journal of the optical Society of America, vol. 14, pp. 1724-1733, 1997.

(Chen 1995) Q. Chen, H. Wu and M. Yachida, "Face Detection by Fuzzy Matching", *Proceeding 5th IEEE international Conference Computer Vision*, pp. 591-596, 1995.

(Chen 2009) Y.W. Chen, R. Xu and A. Ushikome, "Serially-Connected dual 2D PCA for Efficient Face Representation and Face Recognition", *International Journal of Innovative Computing, Information and Control ICIC International*, vol.5, no.11(B), 2009.

(Chin-Chin 1996) H. Chin-Chin, Y. Shin-Ichi, W. H. Fujika and S. Koichiro, "A Fuzzy Self-tuning Parallel Genetic Algorithm for Optimization", *Computers and industrial engineering v*ol. 30, no. 4, pp. 883-893, 1996.

(Cootes 2001) T. F. Cootes, C. J. Taylor, D. Cooper and J. Graham, "Active shape models—their training and application", *Computer Vision and Image Understanding,* vol. 61, no. 1, pp. 18–23, 1995.

(Cootes 2001) T. F. Cootes, G. J. Edwards and C. J. Taylor, "Active Appearance Models", *IEEE Transaction Pattern Analysis Machine Intelligence,* vol. 23, pp. 681–685, 2001.

(Cox 1996) I. J. Cox, J. Ghosn, P. N. Yianilos, "Feature-based face recognition using mixture-distance," *IEEE Computer society conference on computer vision and pattern recognition*, pp. 209, 1996.

(Daempling 2010) H. Daempling, "Xilinx Device Overview", 2005 http://ece.wpi.edu/~haukex/Xilinx_Tables.pdf (May 7, 2010).

(Dagher 2006) I. Dagher and R. Nachar, "Face Recognition Using IPCA-ICA Algorithm", *IEEE Trans. on Pattern Analysis and Machine Intelligene"*, vol. 28, no. 6, 2006.

(Darwin 1972) C. Darwin, "The Expression of the Emotions in Man and Animals", Jonay Murray, 1972.

(Daugman 1989) J.G. Daugman, "Entropy Reduction and De-correlation in Visual Coding by Oriented Neural Receptive Field", *IEEE Transactions on Biomedical Engineering*, vol. 36, no. 1, pp. 107-114, 1989.

(Daugman 1993) J.G. Daugman, "High Confidence Visual Recognition of Persons by a Test of Statistical Independence", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, pp. 1148-1161, 1993.

(Daugman 1997) J. Daugman, "Face and Gesture Recognition: Overview," *IEEE Transaction pattern analysis and machine intelligence*, vol. 19, no. 7, pp. 675-676, Jul. 1997.

(Delac 2004) K. Delac ,M. Grgic, "A Survey of Biometric Recognition Methods", 46th *International symposium electronics in marine*, ELMAR, June 2004, Zadar, Croatia.

(Deschamps 2006) J. Deschamps, G. Jean and G. Sutter, "Synthesis of Arithmetic Circuits FPGA, ASIC, and Embedded Systems", *John Wiley & Sons*, 2006.

(Draper 2003) B. A. Draper, K. Baek, M. S. Bartlett and J. R. Beveridge, "Recognizing Faces with PCA and ICA," *Computer vision and image understanding*, vol 91, issue 1-2 July, 2003.

(Duda 2001) R. O. Duda, P. E. Hart and D. G. Stork, "Pattern Classification", *John Wiley & Sons*, 2001.

(EDK 2006) Embedded System Tools Reference Manual EDK 8.2i, UG111, 2006.

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

(Ellis 1986) H. D. Ellis "Introduction to Aspects of Face Processing: Ten Questions in Need of Answers", *Aspects of Face Processing*, Netherlands, 3–13, 1986.

(Face 2009) The Database of Faces at a glance, 2009.

(Fatemi 2003) H. Fatemi, R. Kleihorst, H. Corporaal and P. Jonker," Real-Time Face Recognition On A Smart Camera", *Proceedings of Acivs (Advanced Concepts for Intelligent Vision Systems), Ghent, Belgium,* pp. 2-5, 2003.

(Field 1994) D.J. Field, "What is the goal of sensory coding", *Neural Computation*, vol.6, pp. 559—601, 1994.

(Shaknarovick 2004) G. Shaknarovick and B. Moghaddam, "Face Recognition in Subspaces", *Mitsubishi Electric Research Laboratory*, training report pp. 2004-041, 2004.

(Galton 1888) F. Galton, "Personal Identification and Description", Nature, pp. 173-188, 1888.

(Gao 2007) W. Gao, B. Cao and S. Shan, D. Zhou, X. Zhang and D. Zhao, "The CAS-PEAL Large Scale Chinese Face Database and Baseline Evaluations," *IEEE Transactions on systems man, and cybernetics,* Part A, vol. 38, no. 1, pp. 149–161, 2007.

(Gavish 2006) B. Gavish, and S. Sridhar, "Computing the 2-median on Tree Networks in *O (n log n)* time", *An international journal of networks,* 2006.

(Giraud 2003) L. Giraud, J. Langou, and M.Rozloznik, "On the Loss of Orthogonality in the Gram-Schmidt Orthogonalization Process", *Technical Report TR/PA/03/25, CERFACS*, 2003.

(Golub 1996) G.H. Golub, C.F. Van Loan, "Matrix Computations", 3rd ed., John *Hopkins University Press*, Baltimore, 1996.

(Gordon 1991) G. GORDON, "Face Recognition Based on Depth Maps and Surface Curvature", *SPIE Proceedings, Geometric Methods in Computer, vol. 1570,* 1991.

(Kanade 1977) T. Kanade, "Computer Recognition of Human Faces", *Interdisciplinary Systems Research*, vol. 47, 1977.

(Kim 2010) H. H. J. Kin, "Survey Paper: Face Detection and Face Recognition", http://www.cs.usask.ca/grads/hyk564/homePage/819/8.

(Haifeng 2007) H. Haifeng, "Orthogonal Neighborhood Preserving Discriminant Analysis for Face Recognition," *Journal of Pattern Recognition Society*, vol. 41, no. 6, pp. 2045-2054, 2007.

(Hallinan 1991) P. W. Hallinan, "A Low-Dimensional Representation of Human Faces for Arbitrary Lighting Conditions", *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition,* pp. 995–999, 1994.

(He 2005) X. He, S. Yan, Y. Hu, P. Niyogi and H. Zhang, "Face recognition using laplacian faces", *IEEE Transaction on pattern analysis and machine intelligence*. vol. 27, no. 3, pp. 328-340, 2005.

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

(Huang 2009) L. Huang, H. Zhuang and S. Morgera, "New Face-Recognition Technology Could Help Security Systems," *IEEE computer society,* 2009.

(IBM 2001) IBM 64-bit processor local bus architecture specification version 3.5, *patent no. SA-14-2534-01*, 2001.

(IBM 2006) IBM product overview PowerPC 405 CPU core, 2006.

(Juell 1996) P. Juell and R. Marsh, "A Hierarchical Neural Network for Human Face Detection", *Pattern Recognition*, vol. 29, no 5, pp. 781-787, 1996.

(Kahan 1997) W. Kahan, IEEE Standard 754 for binary floating-point arithmetic, October 1997.

http://www.cs.berkeley.edu/~wkahan/ieee754status/IEEE754.PDF.

(Kalocsai 1994) P. Kalocsai, I. Biederman and E. E. Cooper, "To What Extent Can The Recognition of Unfamiliar Faces be Accounted for by a Representation of The Direct Output of Simple Cells", *poster presented at annual meeting of the association for research in vision and ophthalmology, Sarasota, Florida, Cognitive Neuroscience*. 1994

(Kamal 2003) R. Kamal, Embedded Systems Architecture, Programming and Design, *Tata McGraw-Hill*, 2003.

(Karp 1997) A. H. Karp and P. Markstein, "High-Precision Division and Square Root", *ACM Transaction on Mathematical Software*, vol. 23, no. 4, p.p. 561-589, 1997.

(Kay 1993) S. M. Kay, Fundamental of Statistical Signal Processing, Estimation Theory VI, *Prentice-Hall*, 1993.

(Kelly 1970) M. D. Kelly, "Visual Identification of People by Computer", technical report AI-130, Stanford AI Project, Stanford, CA, 1970.

(Kim 2001) H. Kim, "Towards Adaptive Balanced Computing (Abc) Using Reconfigurable Functional Caches (RFCS)," *PhD thesis at lowa state university*, 2001.

(Kim 2002) K. I. Kim, K. Jug and H. J. Kim, "Face Recognition Using Kernel Principal Component Analysis", *IEEE Signal processing letter.* vol. no. 9, no.2, 2002.

(Kirby 1990) M. Kirby and L.Sirovich, "Application of the Karhunen-Loeve Procedure for the Characterization of Human Faces", *IEEE Transaction Pattern Analysis and machine intelligence,* vol 12, no. 1, pp. 103-108, 1990.

(Klein 2005) M. Klein, Applications Engineering, Advanced Products Division, Xilinx, Inc, (2005).

(Krisnamoorthy 2007-a) D. Cai, X. He, J. Han and H.J. Zhang, "Orthogonal Laplacianfaces for Face Recognition," *IEEE Transaction image processing*, vol. 15, no. 11, pp. 3608-3614, 2006.

(Krisnamoorthy 2007-b) R. Krisnamoorthy and R. Bhavani, "Face recognition system based on orthogonal polynomials," *Journal of Applied Sciences*, vol. 7, no. 1, pp. 109-114, 2007.

(Kussul 2006) E. M. Kussul, T. N. Baidyk, D. C. Wunsch, O. Makeyev and A. Martin, "Permutation Coding Technique for Image Recognition Systems", *IEEE Transaction on neural networks,* vol. 17, no. 6,

2006.

(Kwon 2009) T. Kwon and J. Draper, "Floating Point Division and Square Root using a Taylor-series Expansion Algorithm", *Microelectronics Journal*, 2009.

(Lade 1993) M. Lades, J. Vorbruggen, J. Buhmann, J. Lange, C. V.D. Alsburg, R. Wurtz, and W. Konen, "Distortion Invariant Object Recognition in the Dynamic Link Architecture. *IEEE Trans. Computer,* vol.42, 300–311. 1993.

(Lanitis 1995) A. Lanitis, C. J. Taylor, and T. F. Cootes, "Automatic face identification system using flexible appearance models", *Image and Visual Computing,* vol. 13, issue 5, pp. 393-401, 1995.

(Lee 2009) T. Y. Lee and Y. H. Fan, Y. M. Cheng and C. C. Tsai, "Hardware-Software partitioning for embedded multiprocessor FPGA systems", *International Journal of Innovative Computing, Information and Control ICIC International,* vol.5, no.10(A), 2009.

(Leung 1996) T. K. Lenung, M. C. Burl and P. Perona, "Probabilistic Affine Invariants for Recognition", *IEEE Proceeding Conference on Computer Vision and Pattern Recogntion,* pp.198-203, 1996.

(Lewicki 2002) M. Lewicki, "Efficient coding of natural sounds", *Nature neuroscience*, vol.5, pp.356-363, 2002.

(Li 1997) Y. Li and W. Chu, " Implementation of Single Precision Floating Point Square Root on FPGAs", *Proceeding of the 5th IEEE symposium on FPGA-based custom computing Machines*, p.p. 226-232, 1997.

(Li 1999) S. Z. LI, AND J. LU, "Face recognition using the nearest feature line method", *IEEE Trans. Neural Network. 10*, 439–443, 1999.

(Li 2000) J. Li, A. Najmi R.M. Gray, "Image Classification by a Two Dimensional Hidden Markov Model". *IEEE Transactions on Signal Processing* vol. 2, no. 48**,** pp. 517–533, 2000.

(Liao 2000) J.R. Liao, "Real-Time Image Reconstruction for Spiral MRI Using Fixed-Point Calculation", *IEEE Transactions on Medical Imaging*, vol.19, no.7, pp. 690-698, 2000.

(Liao 2007) L. Z. Liao, S.W. Luo, and M. Tian, "Whitenedface: Recognition with PCA and ICA", IEEE *Transaction of Signal Processing Letters.* vol. 14, 2007.

(Lin 1997) S. H. Lin, S. Y. Kung, and L. J. Lin. "Face Recognition/Detection by Probabilistic Decision Based Neural

Network", *IEEE Transaction Neural Network,* vol. 8, pp. 114–132, 1997.

(Liu 2000) C. Liu, and H.Wechsler, "Evolutionary pursuit and its application to face recognition" *IEEE Transaction Pattern Analysis Machine Intelligent,* vol. 22, pp. 570–582, 2000.

(Liu 2001) C. Liu and H. Wechsler, "A shape- and texture-based enhanced fisher classifier for face recognition", *IEEE Transaction Image Process,* vol. 10, pp. 598–608, 2001.

(Lu 2003) X. Lu, "Image Analysis for Face Recognition", Department of Computer Sciences & Engineering Michigan State University. www.face-rec.org/interesting-papers/General/ImAna4FacRcg_lu.pdf

(Luc 2005) G. Luc, L. Julien, R. Miroslav and E. Jasper van den, "Rounding Error Analysis of the Classical Gram-Schmidt Orthogonalization Process," *Numerische Mathematik*, vol. 101, no. 1, 2005.

(Mailloux 2007) G. J. Mailloux, S. Simard, and R. Beguenance, "Implementaion of Division and Square Root using XSG for FPGA-based vector control drives", *International Journal of Electrical and Power Engineering*, vol. 5 Issue 1, pp. 524-529, 2007.

(Manjunath 1992) B. S. Manjnath, R.Chellappa, and C. V. D. Malsburg, "A Feature Based Approach to Face Recognition", *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition*. pp. 373–378, 1992.

(Martinez 2001) M.A. Martinez and A.C. Kak, "PCA versus LDA", *IEEE Transaction Pattern Analysis and Machine Intelligence,* vol. 23, no. 2, pp. 228-233, 2001.

(Marwedel 2006) P. Marwedel, Embedded System Design, *Springer*, 2006.

(Memik 2003) S. O. Memik, "Analysis and FPGA Implementation of Image Restoration Under Resource Constraints", *IEEE Transaction on Computer*, vol. 52, no. 3, 2003.

(Middleton 2006) L. Middleton, D. K. Wagg, A. I. Bazin, J. N. Carter and M. S. Nixon, "Developing a non-intrusive biometric environment", *Proceeding IEEE/RSJ, International Conference on Intelligent Robots and Systems,* 2006.

(Ming-Hsuan 2002) Y. Ming-Hsuan, J. K. David and A. Narendra, "Detection Faces in Images: A survey", *IEEE Transaction on Pattern Analysis and Intelligence,* vol. 24, no. 1, 2002.

(Mittal 2008) N. Mittal, E. Walia, "Face Recognition Using Improved Fast PCA Algorithm," *Proceeding congress Image and Signal Processing (CISP)*, pp. 554 – 558, 2008.

(Moghaddam 1997) B. Moghaddam, and A. Pentland, "Probabilistic Visual Learning for Object Representation", *IEEE transaction on*

pattern analysis and machine intelligence, vol. 19, no.7, pp. 696–710, 1997.

(Nefian 1998) A. V. Nefian and M. H. Hayes, "Hidden Markov Models For Face Recognition", *Proc. Inter. Conf. on Acoustics, Speech and Signal Processing, pp.*2721–2724, 1998.

(Niklas 2007) P. Niklas, W. Franz-Erich, and R. Martin, "Laplace Spectra as Fingerprints for Image Recognition", *Computer Aided Design*, vol. 39, pp. 460-476, 2007.

(Oberstar 2007) E. L. Oberstar, Fixed-point representation & fractional math, 2007.

(ORL 2009) http://people.cs.uchicago.edu/~dinoj/vis/orl/, 2009.

(Okada 1998) K. Okada, J. Steffans, T. Maurer, H. Hong, E. Elagin, H. Neven, and C. V. D. Malsburg, "The Bochum/USC Face Recognition System and how it fared in the FERET Phase III Test", In *Face Recognition: From Theory to Applications*, Springer-Verlag, Berlin, Germany, 186–205, 1998.

(Ortega 1963) J. M. Ortega, **"**An error analysis of householder's method for the symmetric Eigenvalue problem", *Numerische Mathematik*, pp.1-225, 1963.

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

(Paul 1995) S. Paul, J. Gotze, and M. Sauer, "Error Analysis of CORDIC-based JACOBI Algorithms", *IEEE Transactions on computers*, vol. 44, no.7, pp. 947-951, 1995.

(Pellerin 2003) D. Pellerin, S. Thibault, "Practical FPGA Programming in C", *Prentice Hall Press*, 2003.

(Pentland 2000) A. Pentland, T. Choudhury, "Face Recognition for Smart Environments," *Computer*, vol. 33, no. 2, pp. 50-55, 2000.

(Peter 1996) S. Peter, and L. Mirian, "Area and Performance Tradeoffs in Floating-point Divide and Square-root Implementations", *ACM Computing Surveys,* vol. 28, no. 3, 1996.

(Phillips 1998) P. J. Phillips, "Support Vector Machines Applied to Face Recognition", *Adv. Neural Inform. Process. Syst.* vol. 11, pp. 803–809, 1998.

(Phillips 2000) P. J. Phillips,H.Moon, S. A. Rizvi, and P. J. Rauss, "The FERET Evaluation Methodology for Face-Recognition Algorithms", *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 10, pp. 1090–1104, 2000.

(Piromsopa 2001) K. Piromsopa, C. Arporntewan, and P. Chongstitvatana, "An FPGA Implementation of a Fixed-point Square root

Operation", *Proceedings of the international symposium on communications and information technology 2001 (ISCIT 2001),* Thailand, pp. 587.589, 2001.

(PPC 2005) PowerPC instruction set extension guide ISA support for the PowerPC APU Controller in Virtex-4 EDK revision 2, 2005.

(Press 1992) W. H. Press, S. A. Teukolsky and W. T. Vetterling, "Numerical recipes in C: the art of scientific computing, Second Edition.", *Cambridge University Press*, 1992.

(Ray 1998) A. Ray, "A Survey of CORDIC Algorithms for FPGA Based Computers", *Proceeding of the ACM/SIGDA sixth international symposium on field programmable gate array*, p.p.191-200, 1998.

(Ronald 2004) S. Ronald and P.K. Viktor, "Computing Lennard-Jones Potentials and Forces with Reconfigurable Hardware", *In International Conference on Engineering of Reconfigurable Systems and Algorithms*, pp. 284-290, 2004.

(Roweis 2000) S. T. Roweis and L. K. Saul "Nonlinear Dimensionality Reduction by Locally Linear Embedding", *Science magazine*, vol. 290, 2000.

(Saul 2003) L.K. Saul and S.T. Roweis, "Think Globally, Fit Locally: Unsupervised Learning of Low Dimensional Manifolds," Journal of Machine Learning Research, vol. 4, pp. 119-155, 2003.

(Rowley 1998) H. Rowley, S. Baluja and T. Kanade, "Neural Network-Based Face Dectection", IEEE Transaction Pattern Analysis and Machine Intelligence, vol. 23-38, 1998.

(Rzempoluch 1998) E. J. Rzempoluch, "Neural Network Data Analysis Using Simulnet", *Spring*, pp. 195-201, 1998.

(Saha 2007) P. Saha, "Automatic Software Hardware Co-design for Reconfigurable Computing Systems", 17th inter. Conference on FPL, 2007.

(Sajid 2008-a) I. Sajid, M. M. Ahmed, I. Taj, M. Humayun and F. Hameed, "Design of High Performance FPGA Based Face Recognition System", *PIERS 2008 in Cambridge, USA, 2-6*, 2008.

(Sajid 2008-b) I. Sajid, M.M. Ahmed and I. Taj, "Design and Implementation of a Face Recognition System Using Fast PCA," *IEEE Proceeding at the international symposium on computer science and its applications (CSA)*, pp.126-130, 2008.

(Sajid 2009-a) I. Sajid, M.M. Ahmed and I. Taj, 2008, "Time Efficient Face Recognition Using Stable Gram-Schmidt Orthonormalization,"

International journal of signal processing, image processing and pattern (IJSIP), vol. 2, no. 1, pp. 35-48, 2009.

(Sajid 2009-b) I. Sajid, M. M. Ahmed and M. Sagheer, "An Improved Face Recognition System Using Stable Orthogonalization in Eigenspace", International Journal of Innovative Computing, Information and Control *(*IJICIC*),* 2010. (Submitted for publication)

(Sajid 2010-a) 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)*, 2010. (Submitted for publication)

(Sajid 2010-b) I. Sajid, M.M. Ahmed and M. Sageer, "FPGA based Optimized Architecture for Face Recognition Using Fixed Point Householder Algorithm", 4th International conference on new trends in information science and service, Gyeongyu, Korea, 2010.

(Samaria 1994) F. Samaria, "Face Recognition Using Hidden Markov Models", *Ph.D. dissertation. University of Cambridge, Cambridge, U.K.*, 1994.

(Samaria 1994a) F. Samaria and S. Young, "HMM based Architecture for Face Identification". *Image Vis. Comput.* vol. *12*, pp. 537–583, 1994.

(Scholkopf 1996) B. Scholkopf, S. Alexander and R. M. Klaus, "Non-linear Component Analysis as a Kernel Eigen Values problem", Neural Computation, vol. 10, no. 5, pp. 1299-1319, 1996.

(Seung 2000) H. S Seung and D. D. Lee, "The Manifold Ways of Perception", *Science*, vol. 290, 2000.

(Shang 2002) L. Shang, A.S. Kaviani and K. Bathala, "Dynamic Power consumption in Virtex-II FPGA family", *Proceedings of the ACM/SIGDA 10th international symposium on field-programmable gate arrays*, pp.157–164, 2002.

(Sharma 2007) A. Sharma, K. K. Paliwal, "Fast Principal Component Analysis Using Fixed-point Algorithm", *Pattern Recognition Letters*, vol. 28, no. 10, pp. 1151-1155, 2007.

(Sim 2003) T. Sim, S. Baker and M. Bsat, "The CMU Pose, Illuminlation, and Expression Database," *IEEE Transaction on pattern analysis and machine intelligence* vol. 25, no.12, pp. 1615–1618, 2003.

(Simon 2008) H. Simon, "A Comprehensive Foundation", *Neural Networks Prentice Hall*, 2008.

(Sirovich 1987) L. Sirovich and M. Kirby, "Low-Dimensional Procedure for the Characterization of Human Face", *Journal of the Optical Society America,* vol. 4, pp. 519–524, 1987.

(Siwabessy 1999) J. Siwabessy, J. Penrose, R. Kloser and D. Fox, , "Seabed habitat classification", *International conference on high resolution surveys in shallow water, sydney,* Australia, 1999.

(Shepherd 1981) J.W. Shepherd, G.M. Davies and H. D. Ellis, "Studies of cue saliency", In *Perceiving and Remembering Faces*, Academic Press, London, U.K. 1981.

(Soderquist 1996) P. Soderquist, Mirian Leeser, "Area and Performance Tradeoffs in Floating-point Divide and Square-root implementations", *ACM computing surveys*, vol. 28, no. 3, 1996.

(Soderquist 1997) P. Soderquist and M. Leeser, "Division and Square Root Choosing the Right Implementation", *IEEE Micro*, pp. 52-66, 1997.

(Sonkamble 2005) S. Sonkamble, R. Thool and B. Sonkamble, "Survey of Biometric Recognition Systems and their Applications", *Journal of theoretical and applied information technology*, vol. 11, no. 1, pp. 45-51, 2005.

(Spacek 2008) L. Spacek, "Collection of facial images: Faces94", 2008. Available: http://cswww.essex.ac.uk/mv/allfaces/

(Stavros 2003) P, Stavros, L. Peter, and B. Miroslaw, "An FPGA System for the High Speed Extraction, Normalization and Classification of Moment Descriptors", *FPL, LNCS 2778, pp. 543–552*. 2003.

(Sung 1997) K. Sung and T. Poggio, "Example-based Learning for View-based Human Face Detection" *IEEE Trans. Pattern Analysis Machine Intelligence,* vol. 20, pp. 39–51, 1997.

(Swets 1996) D. L. Swets and J. Weng, "Discriminant Analysis and Eigenspace Partition Tree for Face and Object Recognition from Views", In *Proceedings, International Conference on Automatic Face and Gesture Recognition*. pp. 192–197, 1996.

(Tanaka 1993) J.W. Tanaka and M. J. Farah, "Parts and Wholes in face Recognition", *Quarterly journal of experimental psychology*, vol. 46, no. 2, 225-245, 1993.

(Tenenbaum 2000) J.B. Tenenbaum, V. de Silva, and J.C. Langford, "A Global Geometric Framework for Nonlinear Dimensionality Reduction," Science, vol. 290, Dec. 2000.

(Thakkar 2006-a) A.J. Thakkar and A. Ejnioui, "Design and implementation of double precision floating point division and square root on FPGAs", *IEEE proceeding,* 2006.

(Thakkar 2006-b) A.J. Thakkar and A. Ejnioui, "Pipelining of double precision floating point division and square root operations", *ACMSE* USA, 2006.

(Toole 2007) A.J.O. Toole, P.J. Phillips, F. Jiang, J. Ayyad, N. Penard and H. Abdi, "Face Recognition Algorithms Surpass Human Matching Faces Over Changes in Illumination," *IEEE Transaction on pattern analysis and machine intelligence*, vol. 27, no. 9, pp. 1642-1646, 2007.

(Turk 1991) M. Turk, A. Pentland, "Eigenfaces for Recognition", *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71-86, 1991.

(Viola 2004) P. Viola and M. J. JONES, "Robust Real-time Face Detection", *International Journal of Computer Vision* vol. 57, no.2, pp.137–154, 2004.

(Wang 2003) X. Wang and B. E. Nelson, "Tradeoffs of Designing Floating point Division and Square root on Virtex FPGAs", *IEEE Proceeding of symposium on field programmable custom computing machines,* FCCM, pp. 195-203, 2003.

(Wei 2004) M. Wei and A. Bigdeli, "Implementation of a real-time automated face recognition system for portable devices", *International symposium on communications and information technologies 2004 (ISCIT 2004),* Japan, 2004.

(Wen 2008) G. Wen, C. Bo, S. Shiguang, C. Delong, Z. Xiaohua, and Z. Debin, "The CAS-PEAL Large-Scale Chinese Face Database and Baseline Evaluations", *IEEE Transaction systems, man, and cybernetics—part a: systems and humans,* vol. 38, no. 1, 2008.

(Wildstart-II) Wildstar-II Hardware Reference Manual revision 5.4, Annapolis Micro system.

(Wilkinson 1962) J. H. Wilkinson, "Error Analysis of Eigenvalue Techniques Based on Orthogonal Transformations", *Journal of the society for industrial and applied mathematics*, vol. 10, no. 1, pp. 162-195, 1962.

(Wilkinson 1965) J. Wilkinson, "The Algebric Eigenvalue Problem", *Oxford university press, london*, 1965.

(Wiskott 1997) L. Wiskott, J. Fellous,. N. Kuiger, and V. Malsburg, "Face Recognition by Elastic Bunch Graph Matching", *IEEE transaction on pattern analysis and machine intelligenc*, vol. 19, issue. 7, pp. 775-779, 1997.

(Xiaofei 2005) H. Xiaofei, Y. Shuicheng, H. Yuxiao, N. Partha and Z. Hong-Jiang, "Face Recognition Using Laplacian Faces," *IEEE transaction on pattern analysis and machine intelligenc*, vol. 27, no. 3, pp. 328-340, 2005.

(Xilinx 2010) Xilinx Virtex II Series FPGA.

(Yale 2002) Yale Univ. Face Database, http://cvc.yale.edu/projects/yalefaces/

(Yamada 2003) S. A. Yamada, A. Nafalski, S.C. Mukhopadhyay, "Dynamic Image Cognition Along With Eigen Patterns", *Application of electromagnetic phenomena in electrical and mechanical systems*, pp. 213-219, 2003.

(Yamin 1997) L. Yamin and C. Wanming, "Implementation of Single Precision Floating Point Square Root on FPGAs", *Proceeding of the 5th IEEE symposium on FPGA-based custom computing Machines,* p.p. 226-232, 1997.

(Yan 2007) S. Yan, D. Xu, and X. Tang, "Face Verification With Balanced Thresholds," *IEEE transaction on image processing*, vol.16, no. 1, pp. 262-268, 2007.

(Yang 2002) M. Yang, D. J. Kriegman, and N. Ahuja, "Detecting Faces in Images: A Survey", *IEEE transactions on pattern analysis and machine intelligence*, vol. 24, no. 1, 2002.

(Yang 2003) Ming-Hsuan Yang, "Extended isomap for pattern classification", *National conference on artificial intelligence*, pp. 224-229, Canada 2002.

(Yuille 1992) A. L. Yuilla, D. S. Cohen and P. W. Hallinan, "Feature Extraction from Faces Using Deformable Templates", *International Journal of Computer Vision,* vol. 8, pp. 99–112, 1992.

(Zaki 2004) G.F. Zaki, R.A. Girgis, W.W. Moussa and W. R. Gobran, "Using HW/SW Co-design to Implement an Embedded Face Recognition/Verification System on an FPGA", report of electronic and communication department Cairo university. http://www.handasarabia.org/mambo/index.php?option=com_docman&task=doc_view&gid=23&lang=Ar

(Zafar 2005) Y. Zafar, "FPGA-compliant Micropipeline Asynchronous Systems", PhD thesis, M.A. Jinnah Univ., 2005.

(Zhang 2005) R. Zhang, h. Chang, "A Literature Survey of Face Recognition and Reconstruction Technique", T*echnical report, university of Texas,* 2005.

(Zhang 2005-b) W. Zhang, S. Shan, W. Gao, X. Chen, and H. Zhang, "Local Gabor Binary Pattern Histogram Sequence (LGBPHS): a Novel Non-statistical Model for Face Representation and Recognition", *Proceeding of the 10th IEEE international conference on computer vision (ICCV '05)*, vol. 1, pp. 786–791, Beijing, China, 2005.

(Zhang 2008) B. Zhang, Z. Wang and B. Zhong, Kernel Learning of Histogram of Local Gabor Phase Patterns for Face Recognition, *EURASIP Journal on Advances in Signal Processing*, 2008.

(Zhao 1998) W. Zhao, R. Chellappa and A. Krishnaswamy, Discriminant Analysis of Principal Components for Face Recognition, *Proceedings of the 3rd. International Conference on Face & Gesture Recognition*, pp. 336, 1998.

(Zhao 2003) W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face Recognition: A Literature Survey," *ACM computing surveys.* vol. 35, no.4, 2003.

(Zhao 2007) D. Zhao, Z. Lin and X. Tang, "Laplacian PCA and its Applications**,"** *Proceeding IEEE 11th international conference on computer vision, ICCV*, pp. 1-8, 2007.