Improved Face Recognition through Infrared Images and Eigenfaces

Developed by

Muhammad Raheel Siddiqui

Supervised by

Prof. Dr. Khalid Rashid
Muhammad Sher

Department of Computer Science
International Islamic University, Islamabad

(2003)

بِسْمِ اللّٰهِ الرَّحْمٰنِ الرَّحِيْمِ

In the name of ALMIGHTY ALLAH,
The most Beneficent, the most
Merciful.

Department of Computer Science,
International Islamic University, Islamabad.

1st November, 2003

Final Approval

It is certified that we have read the thesis, titled "**Improved Face Recognition through Infrared Images and Eigenfaces**" submitted by **Muhammad Raheel Siddiqui** under University Reg. No. **38-CS/MS/01**. It is our judgment that this thesis is of sufficient standard to warrant its acceptance by the International Islamic University, Islamabad, for the Degree of **Master of Science**.

Committee

External Examiner
Dr. Syed Afaq Hussain
Head,
Department of Computer Science,
SZABIST,
Islamabad.

Internal Examiner
Asim Munir
Lecturer,
Department of Computer Science,
Faculty of Applied Sciences,
International Islamic University,
Islamabad.

Supervisors

1. **Prof. Dr. Khalid Rashid**
Dean,
Faculty of Applied Sciences and
Faculty of Management Sciences,
International Islamic University,
Islamabad.

2. **Muhammad Sher**
Assistant Professor,
Department of Computer Science,
Faculty of Applied Sciences,
International Islamic University,
Islamabad.

A dissertation submitted to the
Department of Computer Science,
International Islamic University, Islamabad
as a partial fulfillment of the requirements
for the award of the degree of
Master of Science

Declaration

I hereby declare that this software, neither as a whole nor as a part thereof has been copied out from any source. It is further declared that I have developed this software entirely on the basis of my personal efforts made under the sincere guidance of our teachers. No portion of the work presented in this report has been submitted in support of any application for any other degree or qualification of this or any other university or institute of learning.

Muhammad Raheel Siddiqui
38-CS/MS/01

Dedication

Dedicated to my Family who supported me in all aspects throughout my life.

Acknowledgements

All praise to the Almighty Allah, the most Merciful, the most Gracious, without whose help and blessings, I was unable to complete the project.

Thanks to my Parents who helped me during my most difficult times and it is due to their unexplainable care and love that I am at this position today.

Thanks to my project supervisor Prof. Dr. Khalid Rashid and Muhammad Sher, their sincere efforts helped me to complete my project successfully.

I acknowledge teachers and friends for their help in the project.

Muhammad Raheel Siddiqui

Project in Brief

Project Title:	Improved Face Recognition through Infrared Images and Eigenfaces
Objective:	To Develop an efficient security system for sensitive areas.
Undertaken By:	**Muhammad Raheel Siddiqui**
Supervised By:	**Prof. Dr. Khalid Rashid** Dean, Faculty of Applied Sciences, Faculty of Management Sciences, International Islamic University, Islamabad. **Muhammad Sher** Assistant Professor, Department of Computer Science, International Islamic University, Islamabad.
Technologies Used:	Microsoft® Visual C++ 6.0, MATLAB®
System Used:	Pentium® III
Operating System Used:	Microsoft® Windows® 2000 Professional
Date Started:	1st December, 2002
Date Completed:	30th June, 2003

Abstract

This Thesis report describes an experiment using infrared images and eigenfaces after passing test image through cold effect Enhancement algorithm and sunglasses filtering algorithm. Eigenfaces is one of the most successful techniques used to do automated face recognition. Eigenvectors of the covariance matrix are calculated on the basis of which we create eigenfaces and define eigenspace. The test face is passed through an algorithm to check and enhance if the person come from cold and then is projected to eigenspace to find the match. If match is not found then it is passed through another algorithm to check whether person has worn sunglasses and if so the image is enhanced in order to make recognition more efficient. Different expressions are used to increase the efficiency and check the impact of cold effect enhancement algorithm and sunglasses filtering algorithm. Results from experiments on Infrared face identification and statistical pruning are presented. Two training and two test sets are used for testing. Only one person of the test sets is recognized wrongly with the other but both of them belongs to the training set and it gives 100% accurate results for profile images. Suggested solution gives us satisfactory results in term of speed accuracy and reliability and proves to be competitive software at commercial level. The technique applied in this software can also be used for character Recognition and dimension reduction projects and other fields.

Abbreviations

CSPEC (*P. 5, 12, 16, 20*)	Control Specification
DFD (*P. 5, 7...10, 17*)	Data Flow Diagram
ERD (P. 5, 7)	Entity Relationship Diagram
KL-Transform (*P. 9, 11, 14, 18, 20,* 21, 23, *27, 28, 30, 33, 37, 38, 41, 53, 54, 59*)	Karhnen Loeve Transform
MATLAB (P. 23, 24, 26, 27, 28, 29, 30, 31, 32, 33, 34, 37, *52, 59*)	Matrix Laboratory
PCA (*P. 8, 23, 27, 28, 30, 37, 53, 54, 55, 59*)	Principal Component Analysis
PDL (*P. 7*)	Program Design Language
PSPEC (*P. 5, 7, 11, 16, 20*)	Process Specification
STD (*P. 5, 13, 20*)	State Transition Diagram

TABLE OF CONTENTS

Ch. No. *Contents* *Page No.*

Chapter 1

Introduction

1. INTRODUCTION

Behind every invention or creation, there exists a need. The need behind the invention of airplanes was to allow people to travel faster. The need behind the invention of electric light was to allow better visibility during dark hours. Still, in this day and age many things are being invented based on certain human needs.

Software engineering works on the same principles. A new human need in the world of computers gives birth to a new computer system every time. Hence, our system is based upon strong human need as well.

The project in context deals with the development of Face Recognition through Infrared Images and Eigenfaces a complete software for Securing an organization, a building or any office using Mat lab and Visual C++. It is a successful approach towards recognizing a person face and granting permission to a person to enter the specific area. It also controls the different expressions of face, facial hair, temperature effect and eyewear.

The future enhancement includes the implementation of automatic alignment of the face image to further increase the efficiency of the project.

Image processing as its name describes is a field used to process the images to:

- Facilitate picture storage and transmission
- Prepare images for display or printing
- Enhance and restore images
- Extract information for images etc.

1.1 Biometric

The biometrics and biometry have been used historically to refer to a field statistical method used to analyze biological and environmental phenomena. The word "biometric" has recently been adopted by the Information Technology sector to mean a field of technology devoted to identification of individuals using biological and or behavioral traits. This book will focus on this stream of biometrics.

Biometric methods used to identify people and animals have been used for thousands of years. Subconsciously we are able to differentiate our brothers and sisters by looking at their faces and recognizing subtle differences. Our brains are able to remember those people we have met before and differentiate them from strangers. Our brains are so adapted to this process that we are able to uniquely identify a lost friend in a football stadium given enough time and patience. There is a medical condition for those people who are unable to remember the faces of those they meet; this condition is referred to as prosopagnosia. The ability to differentiate is not limited to people. Only Farmers are able to identify cattle by looking at a cows markings and watching her behave in a herd. However, people who had never seen a cow would find difficulty in identifying them as they would all look the same. To alleviate the problems of being able to identify someone without prior meeting or proving that a business transaction had occurred the Babylonians recorded their fingerprints in clay tablets. Whilst in China the Chinese began to leave fingerprints on important documents [17].

A good Biometric must have Permanence and Individuality.

Some examples of Biometric measurements are:

- Facial Recognition
- Fingerprint
- Iris
- Voice
- Signature
- Keyboard dynamics
- DNA

1.2 Biometric Process

Majority of biometric devices use a five stage process to identify or verify a person's identity.

- *Data Collection:* At first step data is collected, in case of facial recognition the data is the face of the person.
- *Feature Extraction:* Once the data is collected; unique and valuable features are identified to create biometric signature/template.
- *Normalization:* After the signature/template has been created by the features extraction algorithm it may be normalized by another algorithm so that it is in the same format in terms of size and resolution as the rest of the template on the biometric system/ database.
- *Quality Control:* The signature/template is then analyzed to ascertain where or not it is good enough to be used as a comparison signature/template. If it is not good enough to be used then the user is prompted to re-enter his/her biometric sample.
- *Matching:* The normalized template that has been collected is now compared against the enrolled template and its distance is checked to attain a specific threshold.
- *Decision:* Each biometric system has a pre-configurable system policy that sets thresholds or limitations as to what is and what is not an acceptable biometric match. An acceptable match may be based on the quantitative measure and or some other criteria. When the threshold is attained then permission is granted else not.

1.3 Facial Recognition

Facial recognition is an identification or verification of a person looking unique characteristics of his/her face. Face characteristics include shape, color and eye movement and in case of infrared image the structure of veins and tissues. Vendors claim facial recognition systems are intelligent enough to cope with changes in hair style, facial hair, sun tan, glasses and age. They also claim that their systems are intelligent enough to work in areas such as football stadiums, shopping malls and airports. Facial recognition systems scan a predetermined area using video cameras. The video signals produced by these cameras is converted into a series of digital pictures and feed in to the facial detection software. The facial detection software is based on four steps. The facial detection software locates the person and then the face in each digital picture that is fed to it. If a face is found then in case of visible light image background is removed (set to a single color). After the face has been detected and the background has been removed the facial image is normalized. Last step involves comparing the normalized frame against pictures stored in the database. If a face is close to the face stored in the database then the facial recognition software will make a decision based on preconfigured threshold [17].

1.4 The Project

A large no of facial recognition software exist in the market and their results are also good then why we need to build a system which is based on infrared images? There is a very strong reason behind it, although the result of the software in the market is good but they are not 100% efficient. The reason is the problem of twin's recognition. There is only one company named Technology Recognition System who claims that its software is good enough to identify the twins but nothing is available to verify their claim. Technology Recognition System uses the template technique to verify the face images. We apply the Principal Component Analysis technique which is faster as compared to templates. And infrared images are needed because science theory says that the structure of veins and tissues of a person is unique. The use of infrared images can help us in identifying the twins which is not possible using visible light images. In the past, work on the same project has been done, but they have not controlled different other aspects like facial hair, eyewear, and temperature effect.

1.5 Project Scope

Previously the work is done on the face recognition through infrared images and Eigenfaces by Rose Cutler [7]. He has in fact introduced this technique over infrared images before which this technique is used on visible light images but the performance of this technique is very much affected by the noise and its efficiency is almost 60%. Rose Cutler used this technique over infrared images and got accuracy of 96%. He has used only two expressions of the face and did not handle eye wear, facial hair and temperature. His work is not much reliable but a good begin in the field of facial recognition.

The project provides efficient recognition of facial images taken by infrared camera. Previously in 1996 Rose Cutler has done some work on this project. He has used Principal Component Analysis technique on Infrared images but limit the work only to two expressions.

This Project includes the following features.

- Increase the accuracy
- Handle different expressions of face
- Temperature of face of a person
- Facial hair
- Eye wear

1.6 Objectives

The objectives of this project are to develop such a security system based on facial recognition using infrared images and Eigenfaces which will give us maximum accuracy to be used at commercial level and handle different parameters and temperature. This software helps us to reduce false acceptance and false rejection rate. The second objective is to develop software that can not only compete at University level but also at commercial level.

Chapter 2

System Analysis

2. SYSTEM ANALYSIS

At a technical level, software engineering begins with a series of modeling tasks that lead to a complete specification of requirements and a comprehensive design representation for the software to be built. The Analysis model, actually a set of models, is the first technical representation of a system. Over the years many methods have been proposed for analysis modeling. However two now dominate the analysis modeling landscape. The first, *structured analysis* is a classical modeling method and the other approach is *object oriented* method. We have used the first modeling technique for the analysis of the software Face Recognition through Infrared Image and Eigenfaces.

2.1 Structured Analysis

The Structured Analysis Model must achieve three primary objectives.

- Describe what the customer requires.
- Establish a basis for the creation of a software design.
- Define a set of requirements that can be validated once the software is built.

To accomplish these objectives, the analysis model derived during the structured analysis takes the form illustrated in Figure 2.1 [8]. At the core of the model lies the data dictionary — a repository that contains description of all data objects consumed or produced by the software. Three different diagrams surround the core. The entity-relationship diagram (ERD) depicts relationships between data objects. The ERD is the notation that is used to conduct the data modeling activity. The attributes of each data object noted in the ERD can be described using a data object description. The Data flow Diagram (DFD) Provide an indication of how data are transformed as they move through the system and depict the functions and subfunctions that transform the data flow.

The DFD provides additional information that is used during the analysis of the information domain and serves as a basis for the modeling of function. A description of each function presented in the DFD is contained in a process specification (PSPEC).

The State-transition diagram (STD) indicates how the system behaves as a consequence of external events. To accomplish this, the STD represents the various modes of behavioral modeling. Additional information about control aspects of the software is contained in the control specification (CSPEC) [8].

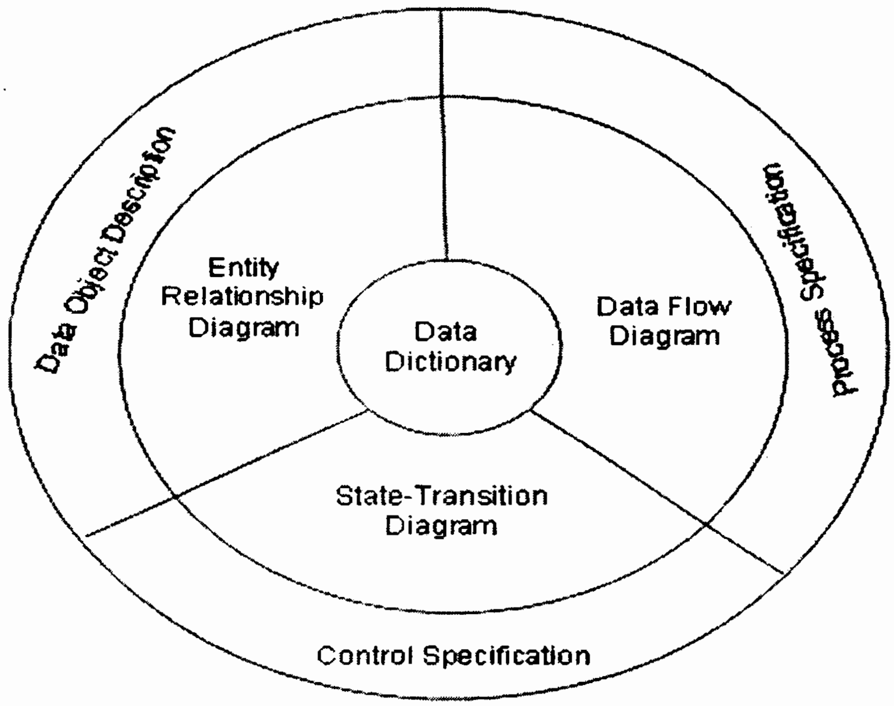


Figure 2.1 Analysis Model

2.1.1 Object Description

In Object Description Section we describe the Objects. Object is a representation. The ImageDBInformation Object is shown in Figure 2.2. Where

- *Image Width* is the width of each image in the Database.
- *Image Height* is the height of each image in the Database.
- *No. of Expressions* is the no of expressions for each subject in our training set.
- *No. of Subjects* is the no of Subjects or person used to create our training set.

- *Subject name file path* is the path to a .txt file which contains the names of the subject sort in order same as of training set.

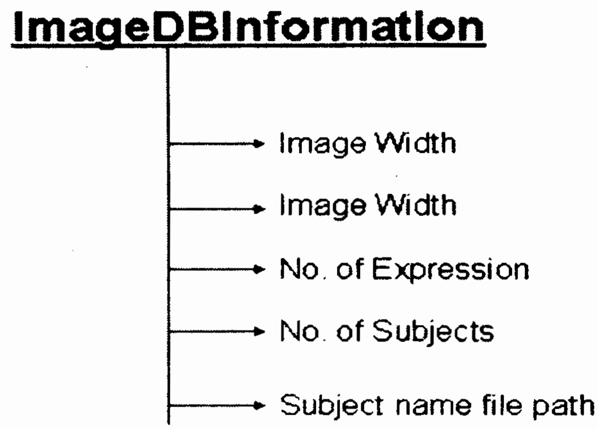


Figure 2.2 ImageDBInformation Object

There is only one object in our software therefore we can not make the ERD of the software.

2.1.2 Data Flow Diagrams (DFD)

As information moves through the software, it is modified by a series of transformations. A DFD is a graphical technique that depicts information flow and the transforms that are applied as data move from input to output. The DFD is also known as *Data flow graph* or a *bubble chart*. Our software is expanded up to the second level DFD and all the functions shown in the DFDs are described in PSPEC.

Figure 2.3 shows the Context level DFD for the software Face Recognition through infrared images and Eigenfaces. This level is the highest level of abstraction where no details are shown only the input to the software and output from software is shown. There is only one bubble which is the software and reveals no function of the software.

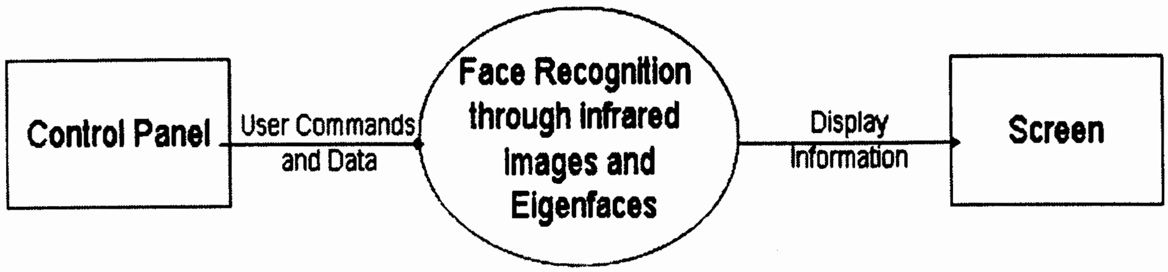


Figure 2.3 Level 0 DFD for Face Recognition through Infrared Images and Eigenfaces

Now the DFD is expanded and level one shows the detail of the process or functions of the software. This level is called level 1 DFD for the software and reveals the function but not the sub function.

Figure 2.4 is the expansion of bubbles in level 0 DFD and here the level of abstraction decreases but only up to the functions still the sub functions are not reveled. It also shows the data storage and the arrow in show which process store the data and which process use the stored data.

The Control Panel is basically the input are or the interface through which user interact with the software and give command for performing PCA, Recognition or reconstruction process. While the Screen is the output device, in our case Screen is the monitor screen and shows the operation status and other proper messages at proper time.

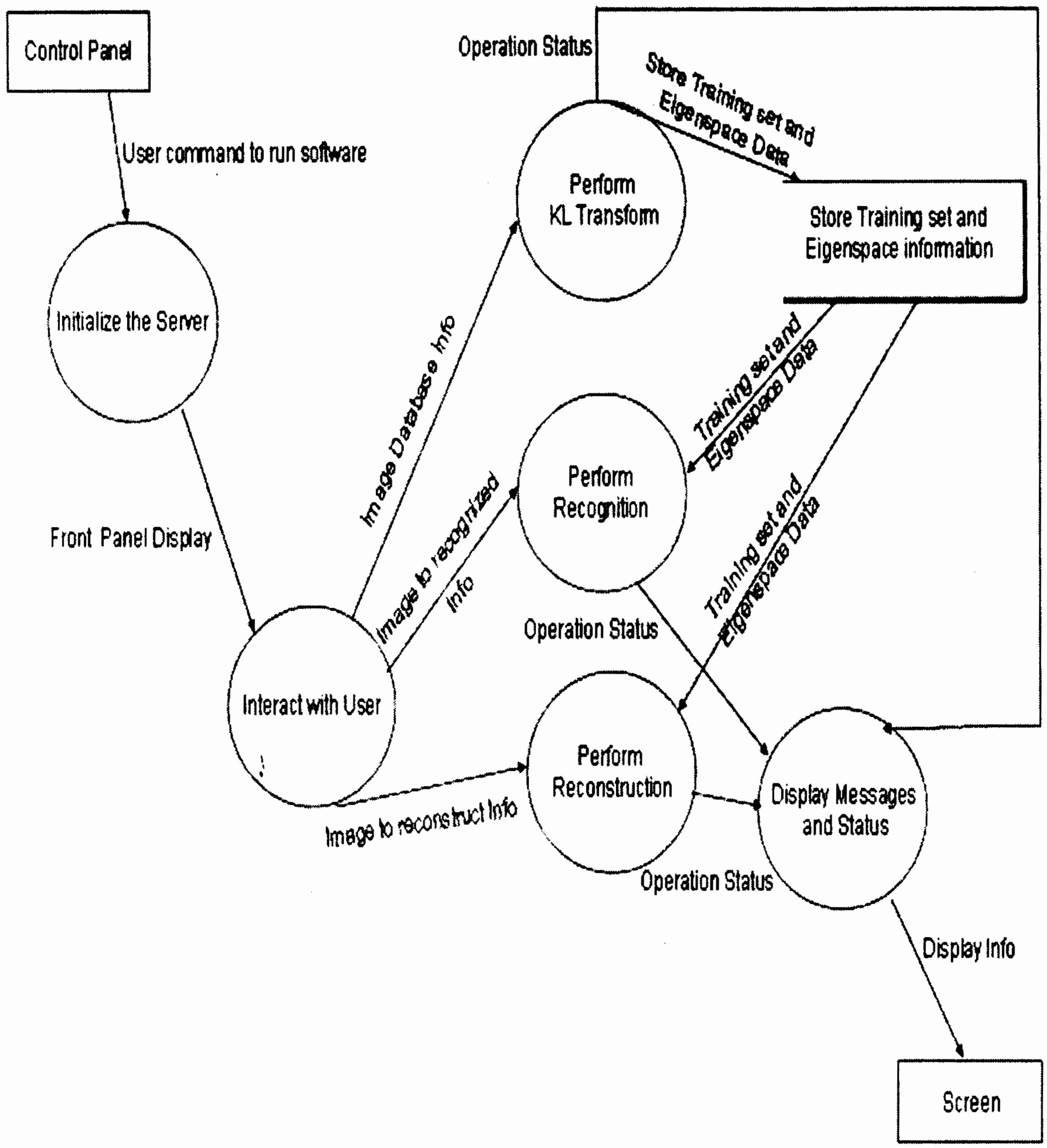


Figure 2.4 Level 1 DFD of Face Recognition through Infrared Images and Eigenfaces

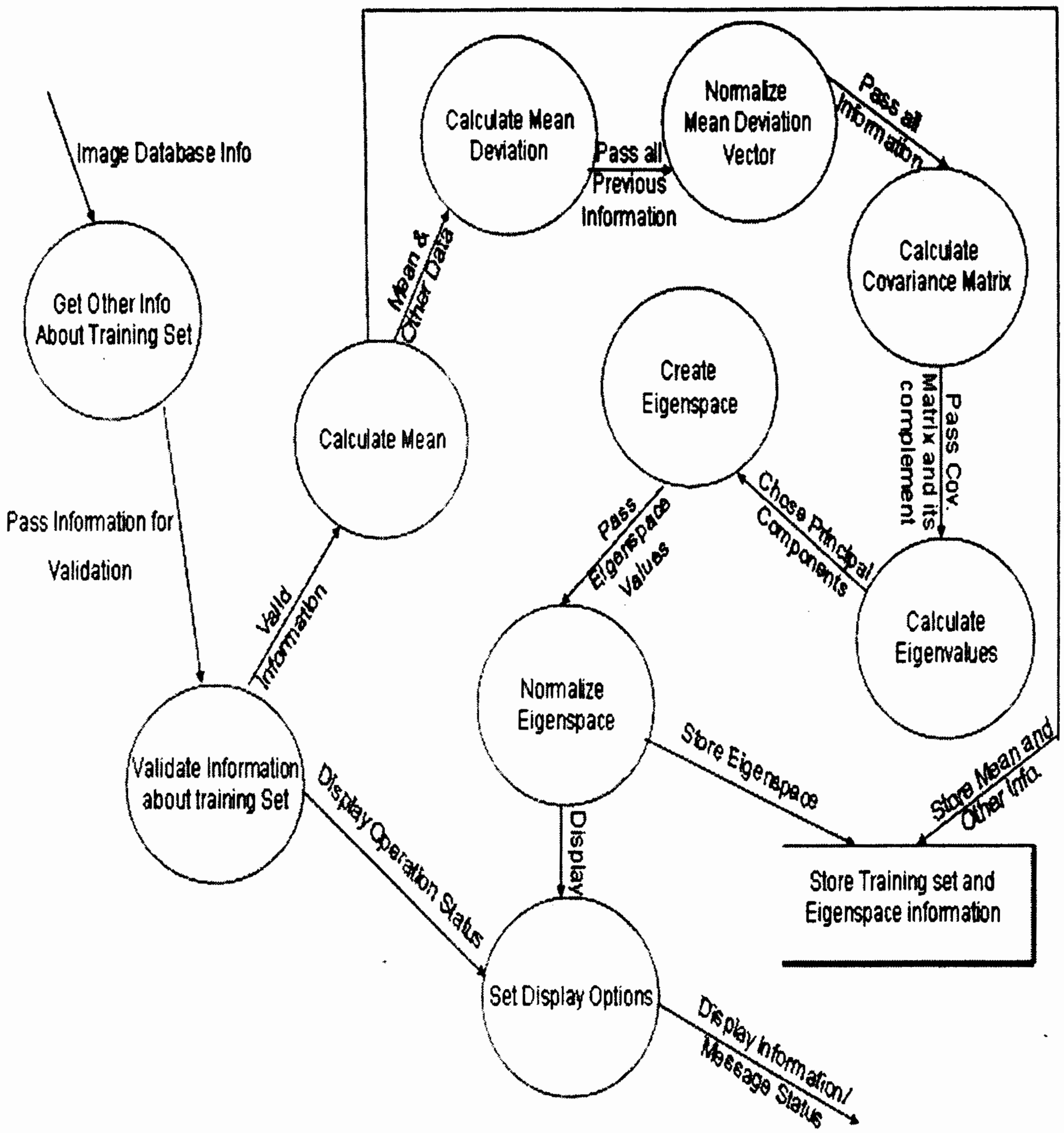


Figure 2.5 Level 2 DFD of KL-Transform Process

Level 2 DFD for KL-Transform process is shown in Figure 2.5. This level shows the sub functions of the KL-Transform process and describes almost all the functionality or calculation involved in KL-Transform Process. The information about mean and the eigenspace is stored after successful creation of the eigenspace. The important thing to note from the whole process of KL-Transform is that the images are not required once the eigenspace is created; only the mean image and the eigenspace are required for recognition.

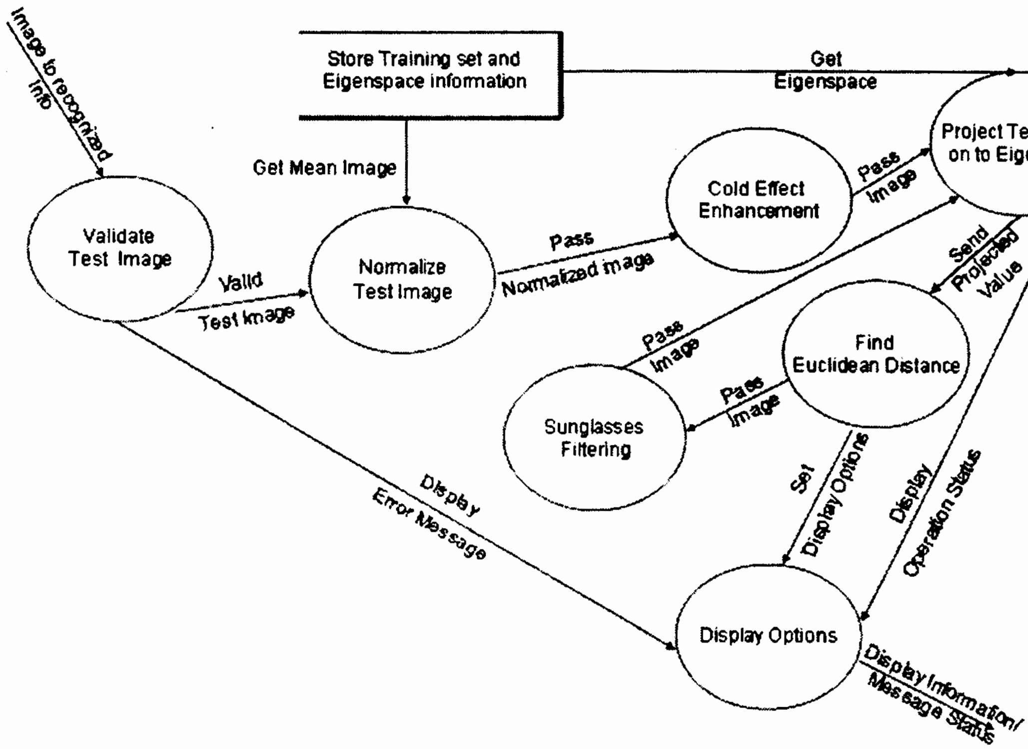


Figure 2.6 Level 2 DFD of Recognition Process

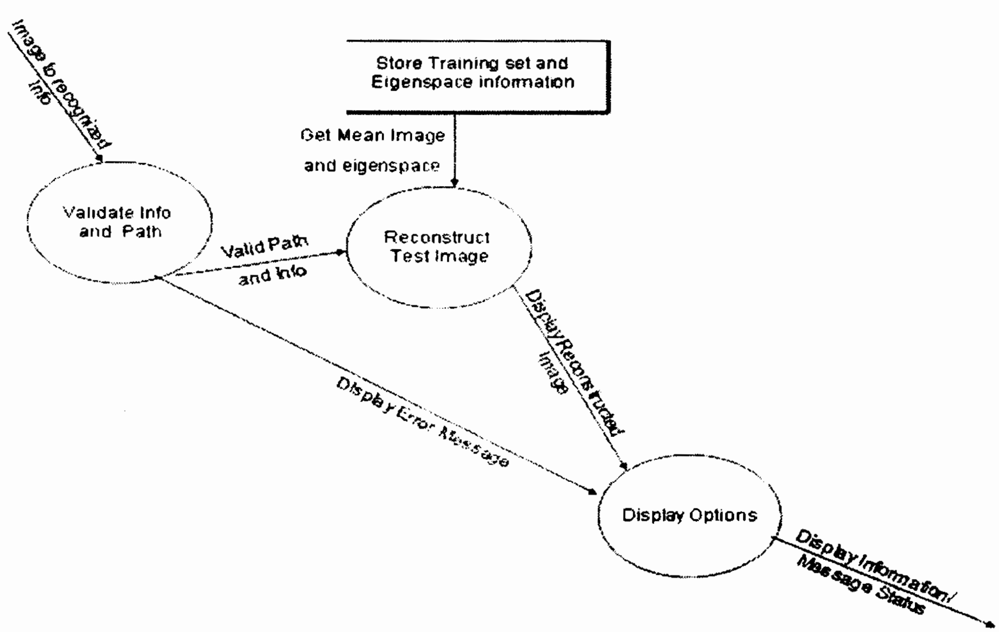


Figure 2.7 Level 2 DFD of Reconstruction Process

The important thing to note in Recognition process described in Figure 2.6 is normalizing Test Image. If the test image is projected to the eigenspace without normalizing then it would produce the wrong results and is one of the major causes in software failure by increasing False Acceptance and False Rejection Rate.

Figure 2.8 describe the reconstruction process. Reconstructed image is not exactly the same as used in the training set because we reduce the dimensions and chose the highest M corresponding eigenvectos which are above the threshold.

2.1.3 Process Specification (PSPEC)

The PSPEC of our software in the form of PDL are:

2.1.3.1 Initialize the Server

Procedure Initialize the server;

Read the command;

If command is valid **then** initialize the server;

Else do not initialize the server;

End if;

Endproc

2.1.3.2 Interact with user

Procedure Interact with user;

Read the input data;

If data is valid **then** send user command to the server;

Else do not send user command to server;

End if;

Endproc

2.1.3.3 Perform KL-Transform

Procedure Perform KL-Transform;

Get other information about the training set and validate it;

If Other Information is not valid **then** print error message;

Else begin

Read the Display Options;

Display the Process Status;

Calculate the mean;

Calculate Mean Deviation of all the images in training set;

Normalize Mean Deviation Vector;

Calculate the Covariance Matrix;

Calculate the Complement of Covariance Matrix;

Calculate the eigenvectors and eigenvalues;

Find the principal component need to make the eigenspace and addition of the eigenvalues is above defined threshold;

Create eigenspace from chosen principal components;

Normalize Eigenspace;

Save Training Set Information, Mean Image and Eigenspace;

End Displaying Process Status;

Display user required results based on display options;

Display the completion message;

End;

End if;

Endproc

2.1.3.4 Perform Recognition

Procedure Recognition;

Get Information about the test image;

Validate the test image;

If validation fails **then** print error message;

```
Else Begin
    Read the Test Image;
    Display Process Status;
    Read Mean Image and Eigenspace constructed from training set;
    Normalize Test Image;
    Project the test image onto the eigenspace;
    Check whether it is a face or not;
    If not a face then Display message not a face;
    Else Begin
        Pass Image to Cold Effect Enhancement Algorithm;
        Calculate the Euclidean Distance of test image from each
        point in eigenspace;
        Match with the defined threshold;
        If distance is below the defined threshold then
            Begin
                Reconstruct the matched image;
                End Displaying Process Status;
                Display user required results based on display
                options;
                Display the completion message;
            End;
        Else Begin
            Pass to Sunglasses filtering algorithm;
            Calculate the Euclidean Distance of test image from
            each point in eigenspace;
            Match with the defined threshold;
            If distance is above the defined threshold then
                    Display Not In Database;
            Else Begin
                Reconstruct the matched image;
                End Displaying Process Status;
```

```
                                Display user required results based on display
                                options;
                                Display the completion message;
                                End;
                            Endif;
                        End;
                    Endif;
                End;
            Endif;
        End;
    Endif;
Endproc
```

2.1.3.5 Perform Reconstruction

```
Procedure Reconstruction;
    Get Image name to be reconstructed;
    Check for the required files;
    If required files not found then print error message;
    Else begin
        Display the Process Status;
        Reconstruct the Image;
        Display Original and Reconstructed Image;
        End Displaying the Process Status;
      End;
    End if;
Endproc
```

2.1.3.6 Display Message and Status

```
Procedure Display Message and Status;
    Get the Message from any process;
```

Set the required Icon;

Display the Message;

End if;

Endproc

2.1.4 State Transition Diagram (STD)

The State Transition Diagram indicates how the system behaves as consequence

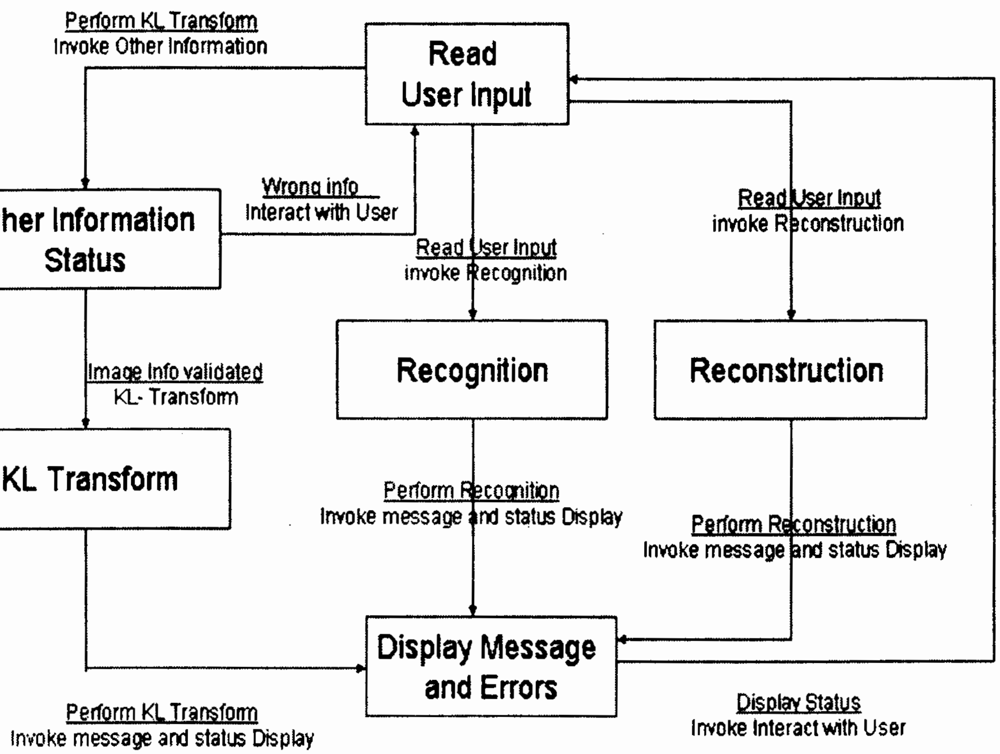


Figure 2.8 State Transition Diagram

of external events. By studying STD, a software engineer can determine the behavior of the system and can ascertain whether there are "holes" in the specified behavior. Figure 2.8 shows the State Transition Diagram for the software Face Recognition thorough Infrared Images and Eigenfaces.

2.1.5 Control Specification (CSPEC)

The Control Specification (CSPEC) represents the behavior of the system in two different ways. One is called specification behavior and the second one is the combinational specification. The CSPEC does not provide any information about the inner working of the processes that are activated as a result of this behavior.

2.1.6 Data Dictionary

The Data Dictionary is an organized listing of all data elements that are pertinent to the system, with precise, rigorous definitions. The Data Dictionary of the software is:

Name:	Image Width
Aliases:	none
Where used/how used:	KL-Transform (input), recognition (input), reconstruction (input)
Description:	ImageWidth= "any numeric value containing any no of digits from 1 to 5".

Name:	ImageHeight
Aliases:	none
Where used/how used:	KL-Transform (input), recognition (input), reconstruction (input)
Description:	ImageHeight= "any numeric value containing any no of digits from 1 to 5".

Name:	NoofExpressions
Aliases:	none
Where used/how used:	KL-Transform (input), reconstruction (input)
Description:	NoofExpressions= "any numeric value containing any no of digits from 1 to 3.

Chapter 3

System Design

3. SYSTEM DESIGN

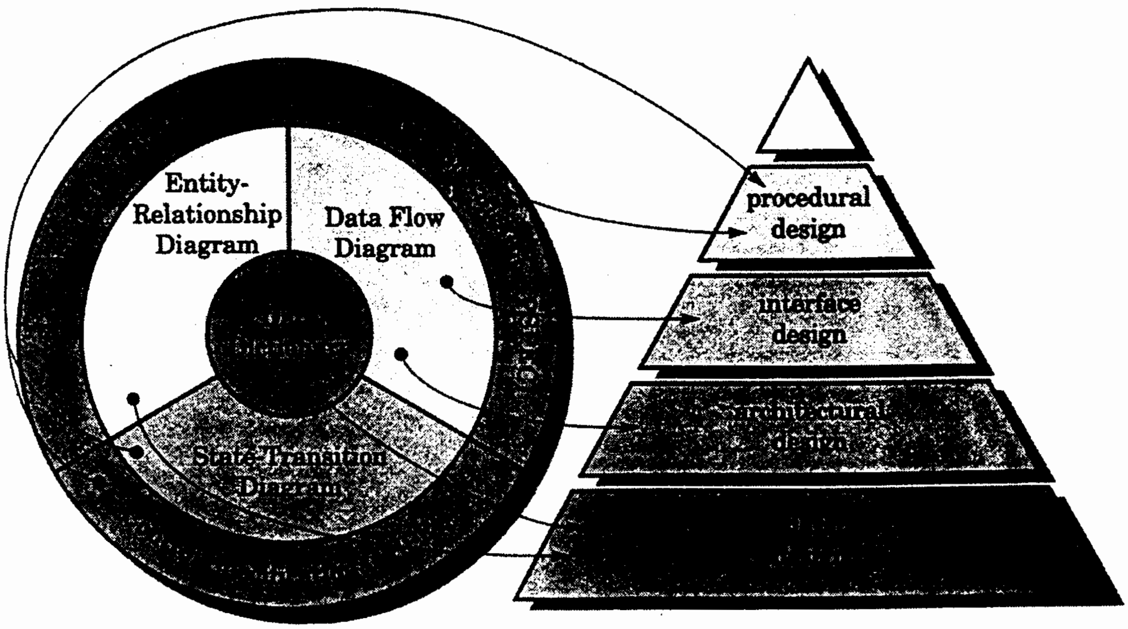
Design is an iterative process transforming requirements into a "blueprint" for constructing the software. It is the first step in the development phase for any engineered product or system. It can also be defined as "the process of applying various techniques and principles for the purpose o defining a device, a process or a a system in sufficient detail to permit its physical realization."

The designer's goal is to produce a model or representation of an entity that will later be built. The process by which the model is developed combines intuition and judgment based on experience in building similar entities, a set of principles and/or heuristics that guide the way in which the model evolves, a ultimately leads to a final design representation [9].

3.1 Relationship of Analysis to Design

For design we need analysis results which serves as base information for design. Infact we explore the analysis in detail and produce desing such that it is directly mapped into coding.

Figure 3.1 [9] shows the realtion of Analysis model to Design model and the arrows shows which of the information from the analysis model is necessary for which design. Data Design is created using the DataDictionary and Entity-Relationship Diagram information of Analysis model. Archicectural Design is creared using the information from Data Flow Diagram of the Analysis model. Interface Design is also created using the infromation from data flow diagram. Procedural Design uses the information from CSPEC. PSPEC and state-transition diagram of the Analysis model.



The analysis model The design model

Figure 3.1 Relation of Analysis model to design model

3.2 Design Types

There are four types of Designs.

- Data Design
- Interface Design
- Architectural Design
- Procedural Design

3.2.1 Data Design

The Data Design transform the information domain model created during analysis into the data structures that will be required to implement the software. The data objects and relationships defined in the entity-relationship diagram and the detailed data content depicted in the data dictionary provide the basis for the data design.

Data Design is the first of four design activities that are conducted during software engineering. The primary activity during data design is to select logical representation of data objects identified during the requirement definition and specification phase. The selection process may involve algorithmic analysis of alternative structures in order to determine the most efficient design or may simply involve the use of a set module that provide the desired operations upon some representation of an object.

As our software contains only one object because it is not a database project and only information about the Training Set so the data design is very simple and contain information about only one object named ImageDBInformation.

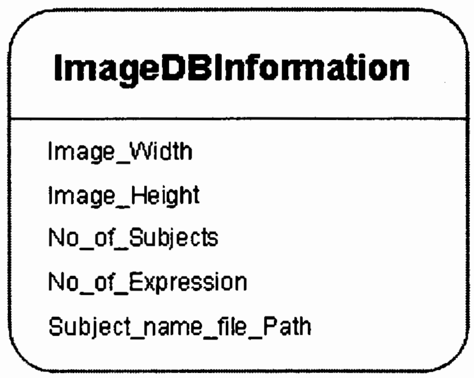


Figure 3.2 ImageDBInformation Object

Figure 3.2 shows the Object ImageDBInformation Object along with its attributes. This information is not stored in a database or in a file. In face this information is stored in registry.

3.2.2 Interface Design

The interface design describes how the software communicates within itself, to systems that interoperate with it, and with humans who use it. An interface implies a flow of information (e.g. data and/or control). Therefore, the data and control flow diagrams provide the information required for interface design. The interface Design can be seen in Appendix A. The entire diagrams related to the interface are shown there.

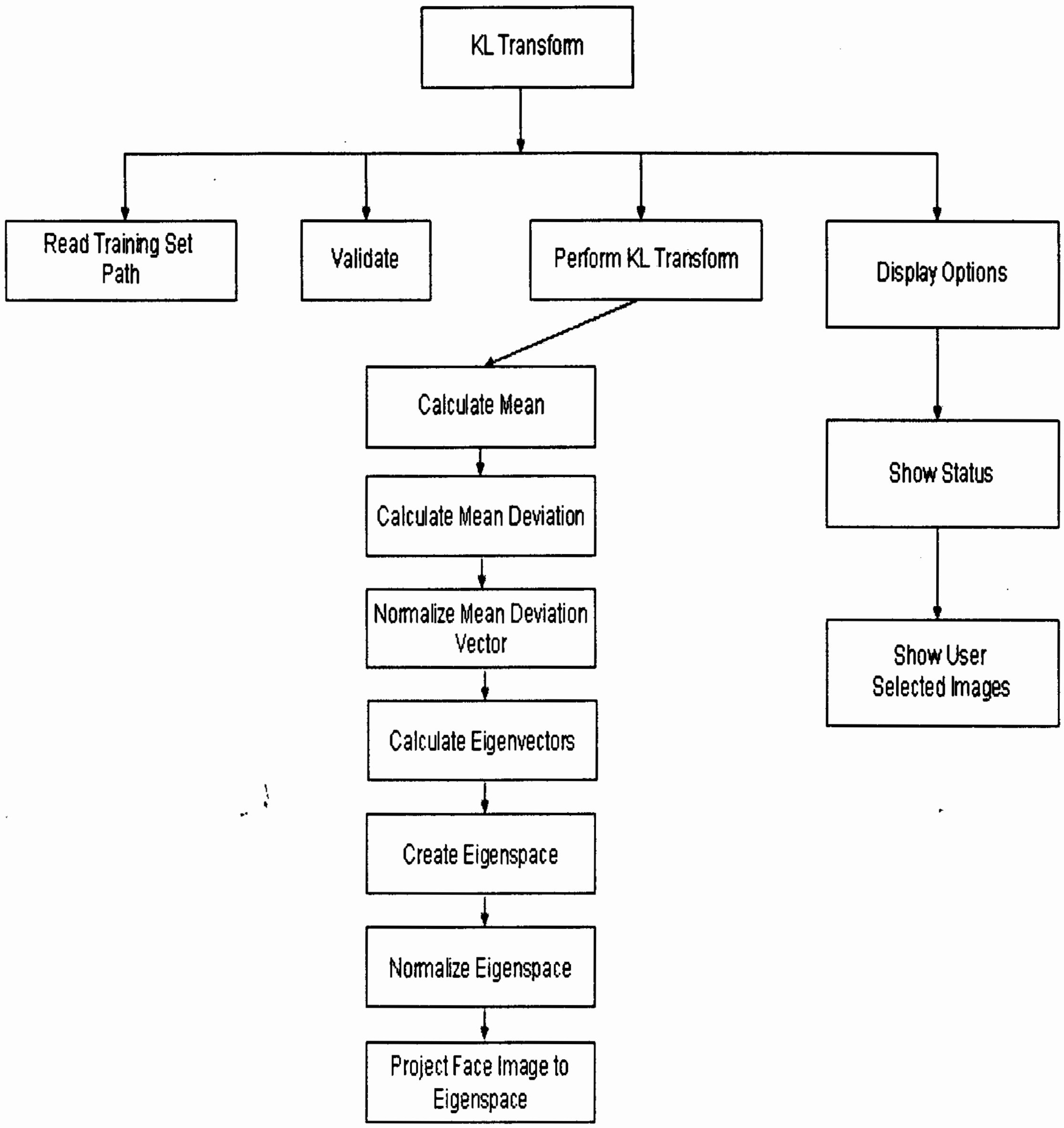


Figure 3.5 Program structure of KL-Transform process

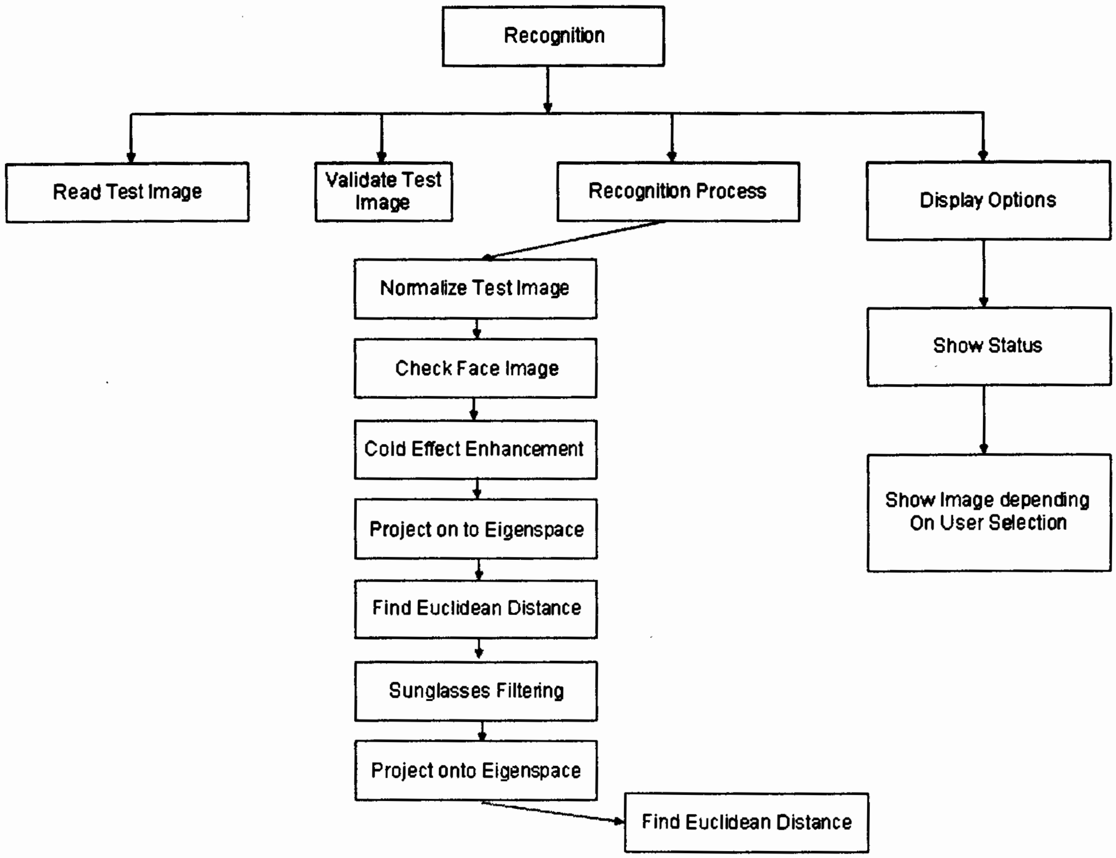


Figure 3.6 Program structure of Recognition process

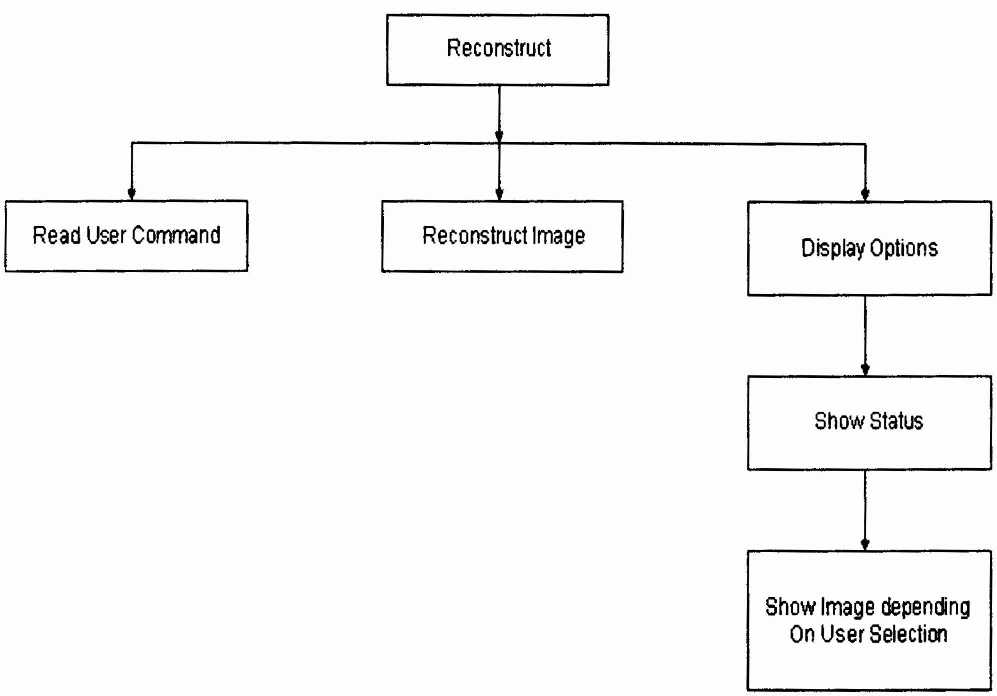


Figure 3.7 Program structure of Reconstruction process

3.2.4 Procedural Design

The procedural design transforms structural elements of the program architecture into a procedural description of software components. Information obtained from the PSPEC, CSPEC and STD serve as the basis for procedural design. The Procedural Design for the software is shown in the Figures 3.8, 3.9, 3.10 and 3.11.

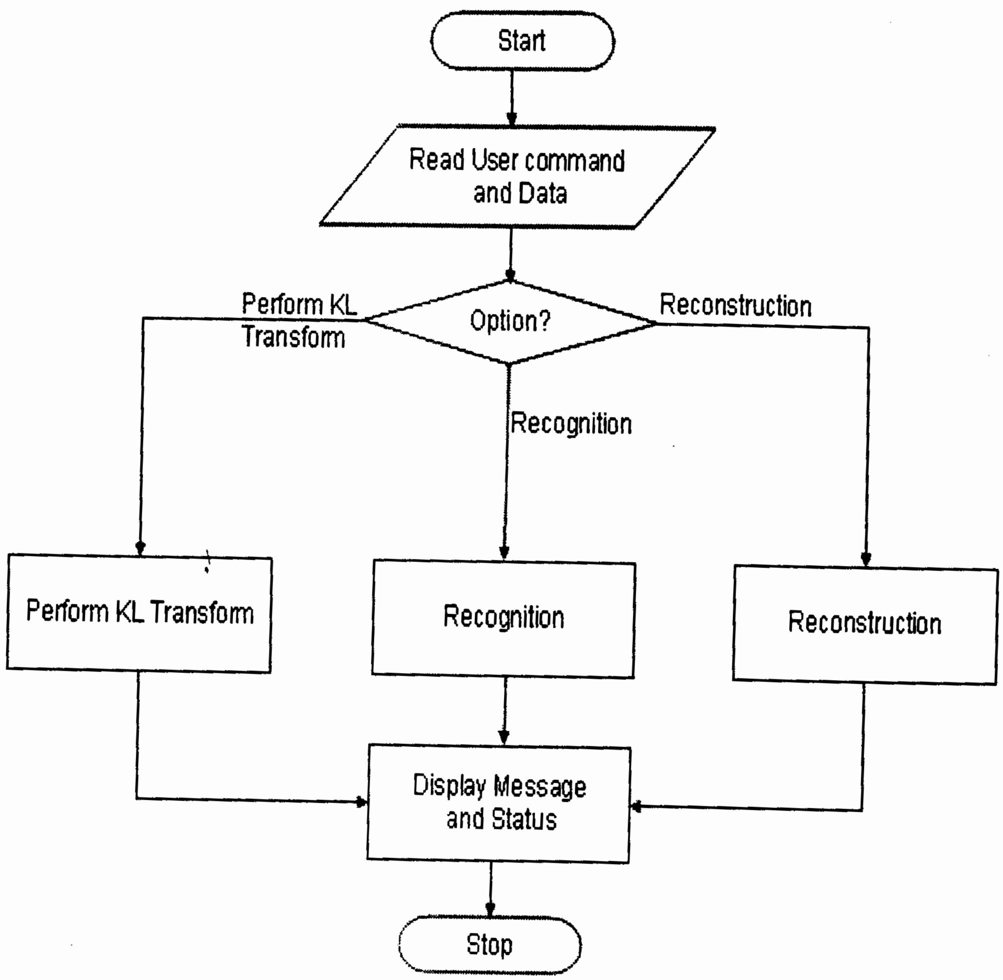


Figure 3.8 Procedural Design

Figure 3.10, 3.11 and 3.12 are the procedural diagrams of KL-Transform, Recognition and Reconstruction processes respectively.

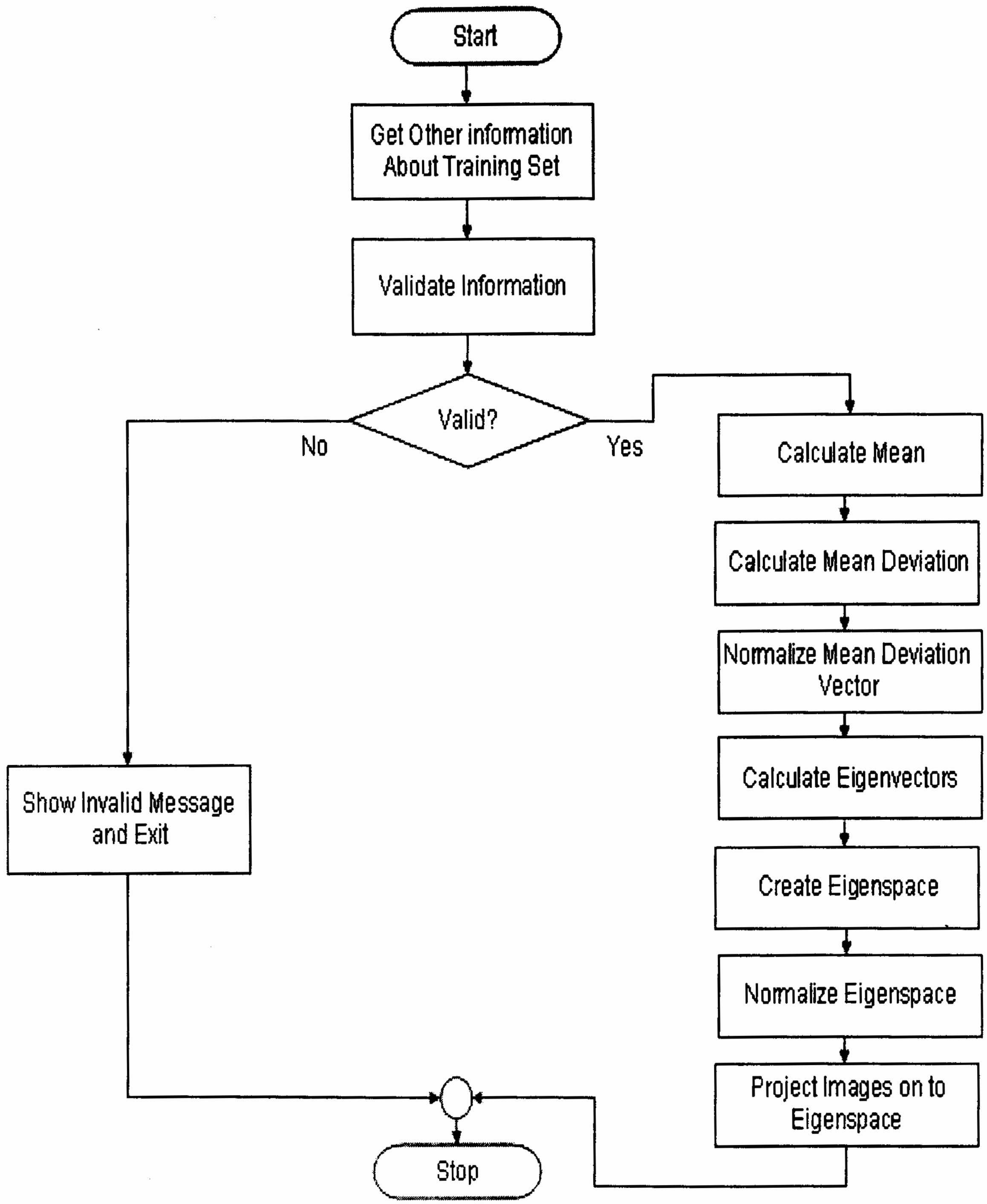


Figure 3.9 Procedural Design of KL-Transform

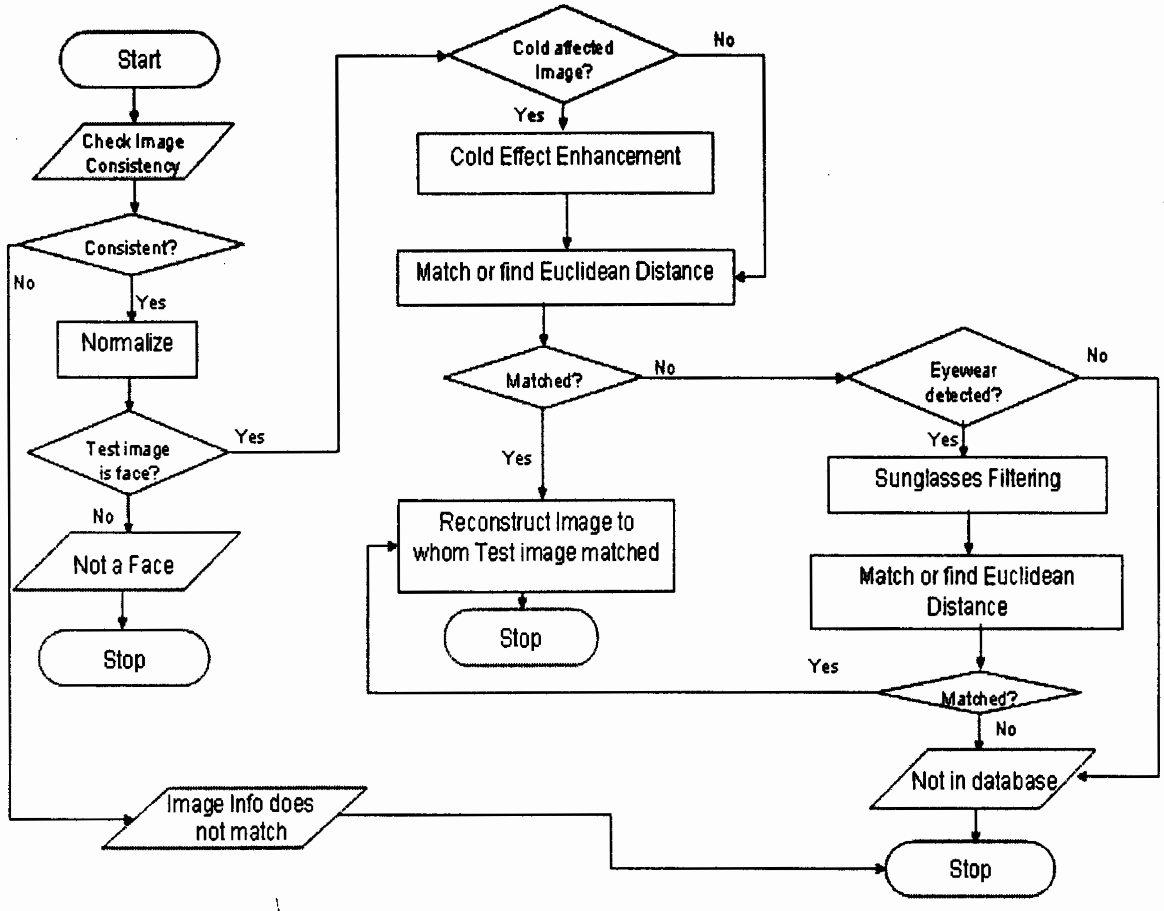


Figure 3.10 Procedural Design of Recognition process

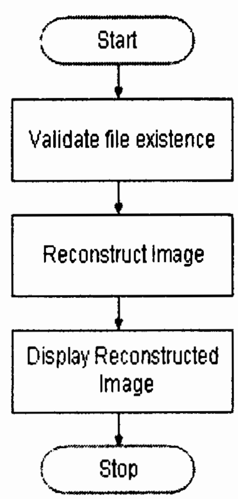


Figure 3.11 Procedural Design of Reconstruction process

The Design Activity is completed here and can easily be mapped to coding in implementation section which is last activity of the software development Process.

Chapter 4

System Development and Implementation

4. SYSTEM DEVELOPMENT AND IMPLEMENTATION

This is the second last activity in the project and comes before testing of the whole program, however the partial testing can be done during implementation after completion of ur every module.

Our software is developed using two tools Visual C++ and MATLAB. And the method used to implement the facial recognition is PCA and is selected cosidering Speed, Accuray and Error Reduction Rate. Our softwares is divided into three modules KL-Transform, Recognition and Reconstruction.

4.1 Client Application and MATLAB Server

Our software works as a Client Server Application and use the non-sharing connection where the application created using Visual C++ works as a clients and handle all the fuctionality realted to Graphical User Interface. While Computational efficiency of MATLAB is used and it works as a server and all the commands related to complex computations are performed by the server.

To Create a Server Client relationship MATLAB Provides a library Using which we can create a server and client type relationship. We can create two type of connection to the server [10].

- Shared Connection
- Non-sharing Connection.

4.1.1 Shared Connection

In this type of connection, if the MATLAB Server is not started by any other application it will start the MATLAB Server in the sharing environment. And when any other application who also has a sharecd connection facility interacts with the Server then it will now not create other instance of MATLAB Server but just directly link it to the previously opened instance. And now these two application can share the MATLAB wrokspace variables in their application.

4.1.2 Non-Sharing Connection

In this type of connection, if the MATLAB Server is not started by any other application it will start the MATLAB Server and when any other application call for MATLAB Server connection then it will create a new instance of MATLAB Server and these two application can not share the MATLAB wrokspace variables.

4.1.3 Library Function Description

The Library functions and their description used to connect to MATLAB Server are:

engClose	Quit engine session
engEvalString	Evaluate expression in string
engGetArray	Copy variable from engine workspace
engGetVisible	Determine visibility of MATLAB engine session
engOpen	Start engine session
engOpenSingleUse	Start engine session for single, nonshared use
engOutputBuffer	Specify buffer for MATLAB output
engPutArray	Put variables into engine workspace
engSetVisible	Show or hide MATLAB engine session

These are the Functions and their description which are used to get variable values using from the MATLAB workspace.

mexAtExit	Register function to be called when MATLAB is cleared or terminates
mexCallMATLAB	Call MATLAB function or user-defined M-file or MEX-file
mexErrMsgTxt	Issue error message and return to MATLAB
mexEvalString	Execute MATLAB command in caller's workspace
mexFunction	Entry point to C MEX-file
mexFunctionName	Name of current MEX-function
mexGet	Get value of Handle Graphics property
mexGetArray	Get copy of variable from another workspace
mexGetArrayPtr	Get read-only pointer to variable from another workspace
mexIsGlobal	True if mxArray has global scope
mexIsLocked	True if MEX-file is locked
mexLock	Lock MEX-file so it cannot be cleared from memory
mexMakeArrayPersistent	Make mxArray persist after MEX-file completes
mexPrintf	ANSI C printf-style output routine
mexPutArray	Copy mxArray from your MEX-file into another workspace
mexSet	Set value of Handle Graphics property
mexSetTrapFlag	Control response of mex Call MATLAB to errors
mexUnlock	Unlock MEX-file so it can be cleared from memory
mexWarnMsgTxt	Issue warning message

4.2 Functionality Added to Software Using Visual C++

All the errors related to the software except computational errors are handled Using Visual C++ and it show all the error messages and other interactive message to the user. Whole Graphical Interface is designed using Visual C++ and Screen Shots of which are provided in User Manual (See Appendices Section). Following functionalities are added to the software using visual C++.

- Splash Screen at the start of the software.
- Single Instance of Application.
- User friendly interface for user ease and Functionality.

Registry Entry of mfiles folder path, Information related to training set and the path of the training set images directory.

Directory Selection Dialog box for selecting the path.

Wait While Performing Operation Dialog Box.

Accelerator Keys

Proper Error and interactive message for the user.

Help file.

4.2.1 Splash Screen

The Splash Screen or Welcome Screen Window is shown during the initialization of the MATLAB Server and it also shows the status of the Application. It is a very good technique to show something on the screen while preprocessing of application is going on.

```
CWelcomeDlg* m_WelcomeDlg;
        m_WelcomeDlg = new CWelcomeDlg();

        m_WelcomeDlg->Create(IDD_WELCOMEWINDOW, this);
        m_WelcomeDlg->ShowWindow(SW_SHOW);
        if(!GetValFromRegistry())
        {
```

```
            .       .       .       .
            .       .       .       .
            .       .       .       .
        }
m_WelcomeDlg->m_Message.SetWindowText("Initializing Matlab Server...");
        UpdateData(false);
        ep=NULL;
        int *tmp;
        tmp=0;
        if(!(ep=engOpenSingleUse("\0",NULL,tmp)))  //Not a shared connection
                AfxMessageBox("Unable to initialize Matlab
Server.",MB_ICONERROR);
        else
        {
            .       .       .       .       .
            .       .       .       .       .
            .       .       .       .       .
        }
        m_WelcomeDlg->m_Message.SetWindowText("Starting Application...");
        UpdateData(false);
        Sleep(1000);
        .       .       .       .
        .       .       .       .
        .       .       .       .

        if(engEvalString(ep."addpath "+mfilesPath+";")!=0)
                AfxMessageBox("Matlab Server not
Initialized",MB_ICONERROR);
        m_WelcomeDlg->DestroyWindow();
        delete m_WelcomeDlg;
```

4.2.2 Single Instance of Application

Only one instance of the Application is allowed to run in order to maintain consistency between the whole recognition process and the eigenspace.

```
pWndPrev=CWnd::FindWindow(NULL,"Face Recognition through Infrared
Images
& Eigenfaces");
        if(pWndPrev!=NULL)
        {
                if(pWndPrev->IsIconic())
                        pWndPrev->ShowWindow(SW_RESTORE);
                else
                        pWndPrev->SetForegroundWindow();

                return FALSE;
        }
```

FindWindow() Function will find the previous instance of the window and if found the application previous instance will popup in front of all other windows.

4.2.3 User Friendly Interface and Functionality

The Main Application Screen is divided into three areas.

KL-Transform (PCA)

Recognition

Reconstruction

4.2.3.1 KL-Transform Interface

This is the first and the most important part of the software KL-Transform creates the actual eigenspace which is the base of the software and before recognition process it is necessary to make the eigenspace. If the user has run the software for the first time then until or unless he creates the eigenspace, he can not recognize or reconstruct the page. It connects with the MATLAB using the library defined above. And also send commands to the MATLAB Server who actually creates the eigenspace. This area defines the following things:

The first Edit Box Labeled Image Database Path and a limit is put of 243 character entry using function

m_DBImagePath.LimitText(243);

Where *m_DBImagePath* is the variable used for this edit box. And its value is initialized if found in the registry using the function

m_DBImagePath.SetWindowText(IMDBPath);

and *IMDBPath* is a protected member to store the Training Set Path

Browse button brings the *Select Image Directory* Dialog in front of the main application and user can select the path from there. When user presses the button following code is executed to perform the operation.

```
    if(SelDir->DoModal()==IDOK && SelDir->PathtoTransfer!="\\")
        m_DBImagePath.SetWindowText(SelDir->PathtoTransfer);
```

Before this code the instance of the Dialog is created. The *DoModal()* shows the dialog infront of the screen and when after selection of valid path user press ok then *SetWindowText()* sets the path to the edit box of the main application window. At the end the instance of the Dialog is deleted.

3. Then there is a Display Section. While performing the operation if user want to see the original images, Mean image and eigenfaces, he can check the desired checkbox and this information is passed to the MATLAB and then MATLAB will show the required figures and it is described under the MATLAB functionality heading.

4. *Perform PCA* is the button which actually perform the KL-Transform but before this it checks for the necessary file existence and if files are not found it will print the error message and the function will not perform and if files are present there then it display a *wait while* window and perform the operation by passing a command string to MATLAB Server which tell Server to perform that function or command passed to it.

```
if(!PathFileExists(Path))
{
```

```
        AfxMessageBox("\""+Path+"\" is not a valid path.",MB_ICONERROR);
        . . . . .
        . . . . .
        return;
}
```

PathFileExists() checks whether or not the given path is a valid path or not it not the it will print the error message and return from the function

```
        if(engEvalString(ep,adddpath)!=0)
        {
        . . . . .
        . . . . .
        AfxMessageBox("Matlab Server not initialized.",MB_ICONERROR);
        return;
        }
```

Adding path to the MATLAB Environment Variable of the required MATLAB files.

```
        CString FuncForML="Reply=klt('";
        .       .       .       .       .
        .       .       .       .       .
        .       .       .       .       .
        FuncForML+="',";
```

Creating the string to be the command send to the MATLAB Server.

```
        if(m_DisOrigionalImage.GetCheck())
                FuncForML+="'y',";
        else
                FuncForML+="'n',";

        if(m_DisMeanImage.GetCheck())
```

```
            FuncForML+="'y',";
        else
            FuncForML+="'n',";

        if(m_DisEigenface.GetCheck())
            FuncForML+="'y'";
        else
            FuncForML+="'n'";
        FuncForML+=");";
```

String has been created which will now send as a command to MATLAB Server

```
        if(engEvalString(ep,FuncForML)!=0)
        {
        .    .    .    .    .
        .    .    .    .    .

        AfxMessageBox("Matlab Server not initialized.",MB_ICONERROR);
        return;
        }
```

The above written function will Send the command to MATLAB Server.

```
        mxArray *result;
        result= engGetArray(ep,"Reply");
        if(result!=NULL)
        {
        double Reply=mxGetScalar(result);
        if(Reply==1)
        {
            FillReconstructionImageList();
            addvaluestoregistry(Path, namefilePath,atoi(Sub),atoi(Exp),
            atoi(Wid));
```

```
            .    .    .    .    .    .
            .    .    .    .    .    .
            AfxMessageBox("KL-Transform(PCA) successfully
            completed.",MB_ICONINFORMATION);
        }
        else if(Reply==0)
            AfxMessageBox("Image don't match with image information
            provided.",MB_ICONERROR);
        else
                AfxMessageBox("Unsuccessful performing KL-
                Transform(PCA).",MB_ICONERROR);
            mxDestroyArray(result); //Release the space of mxArray variable
                                "Result"
        }
        else
        {
            AfxMessageBox("Unsuccessful recognizing due to Invalid
                command to load file.",MB_ICONERROR);
            return;
        }
```

The Above code will send command to MATLAB Server and then get the value return by the command function to check whether or not the function is performed well enough and display the appropriate messages.

4.2.3.2 Recognition Interface

This is the 2nd part of the software and used to show the recognition of infrared face image of the person. This function takes the test image path and then verifies the path and file existence necessary for recognition and the whole part can be described in following points.

1. There is the first Edit Box Labeled *Image to be Recognized* and a limit is put of 255 character entry using function

   ```
   m_RecImagePath.LimitText(255);
   ```

   Where *m_RecImagePath* is the variable used for this edit box.

2. Browse button brings the *Open* Dialog in front of the main application and user can select the path from there. When user presses the button following code is executed to perform the operation.

   ```
   CFileDialog* FileDialog;
   .    .    .    .    .
   .    .    .    .    .
   .    .    .    .    .
   if(FileDialog->DoModal()==IDOK)
   {
       ImagetoRecognizePath=FileDialog->GetPathName();
       m_RecImagePath.SetWindowText(ImagetoRecognizePath);
   }
   delete FileDialog;
   ```

   First the instance of the Dialog is created. The *DoModal()* shows the dialog in front of the screen and when after selection of valid path user press ok then *SetWindowText()* sets the path to the edit box named *m_RecImagePath*. At the end the instance of the Dialog is deleted.

3. Then there is a Display Section. While performing the operation if user want to see the Enhancement Area and Ehanced Image, he can check the desired checkbox and this information is passed to the MATLAB and then MATLAB will show the required figures and it is described under the MATLAB functionality heading. The important thing to remember is that it always shows the Test Image and the Matched Image (if any).

4. *Recognition* is the button which actually perform the Recognition of test image but before this it checks for the necessary file existence and if files are not found

it will print the error message and the function will not perform and if files are present there then it display a *wait while* window and perform the operation by passing a command string to MATLAB Server which tell Server to perform function or command passed to it.

```
if(!PathFileExists(Path))
{
        AfxMessageBox("\""+Path+"\" is not a valid path.",MB_ICONERROR);
        . . . . .
        . . . . .
        return;
}
```

PathFileExists() checks whether or not the given path is a valid path or not it not the it will print the error message and return from the function

```
if(!FileExistence(mfilesPath+"recognition.m") || … ||............)
        return;
```

File Existence of the necessary MATLAB Files is done here

```
CString FuncForML="Reply=recognition('";
.       .       .       .
.       .       .       .
.       .       .       .
FuncForML+="',";
if(m_DisEnhancementArea.GetCheck())
        FuncForML+="'y',";
else
        FuncForML+="'n',";

if(m_DisEnhancedImage.GetCheck())
        FuncForML+="'y'";
```

```
else
        FuncForML+="'n'";
FuncForML+=");";

if(engEvalString(ep,FuncForML)!=0)
{
        AfxMessageBox("Matlab Server Not initialized.",MB_ICONERROR);
        return;
}
```

Above code send the recognition command to the Matlab Server.

```
mxArray *result=NULL;
result= engGetArray(ep,"Reply");
if(result!=NULL)
{
        double Reply=mxGetScalar(result);
        if(Reply==0)
        AfxMessageBox("Image does not match Information provided.\n It
        should be consitent with Training Set Images.",MB_ICONERROR);
        else if(Reply!=1)
                AfxMessageBox("Unsuccessful recognizing due to unexpected
                error.",MB_ICONERROR);
        mxDestroyArray(result); //Release the space of mxArray variable "Result"
}

else
        AfxMessageBox("Unsuccessful recognizing due to invalid command to
                load file.",MB_ICONERROR);
```

The Above code gets the value return by the command function to check whether or not the function is performed well enough and display the appropriate messages.

4.2.3.3 Reconstruction Interface

This is the 3rd part of the software and used for reconstructing face image of the person from eigenspace.. The image name is selected from the list available on the left side; the whole part can be described in following points.

1. There is Combo Box Labeled *Image to be Reconstructed* named *m_ReconImgList* it is filled after the successful performance of the KL-Transform. And the next time software run this list is filled using registry values of subject and expressions.

2. *Reconstruction* button which actually perform the Reconstruction, before this it checks for the necessary file existence and if files are not found it will print the error message and the function will not perform and if files are present there then it display a *wait while* window and perform the operation by passing a command string to MATLAB Server which tell Server to perform function or command passed to it.

```
if(!FileExistence(mfilesPath+"reconstruction.m") || ... ||............)
        return;
```

File Existence of the necessary MATLAB Files is done here

```
CString FuncForML="reconstruction('";
.       .       .       .
.       .       .       .
.       .       .       .
FuncForML+="');";
if(engEvalString(ep,FuncForML)!=0)
{
        .       .       .       .
        .       .       .       .
        AfxMessageBox("Matlab Server not initialized.",MB_ICONERROR);
        return;
```

}

Above code send the reconstruction command to the MATLAB Server. and the result are shown on the screen.

4.2.4 Registry

The Registry is used to store two things. One is the value without which our software will not run so that no one can copy or steel our software and secondly it store the information about the training set so that they can be used the next time software is initiated. Following are the function used to manipulate the Registry.

1. The Instance of Class CRegKey.

   CRegKey RegKey;

2. To Set Registry value

   SetValue(KEY, "SubKEY", “newvalue”, "ValueNametobeSet");

3. To Open Registry Key

   Open(KEY, "SUBKEY");

4. To Retrieve Value From Registry Key

   QueryValue(Buffer, "Valuetobequried", PtrofBufSize)

5. To Close Registry Key

   Close();

4.2.5 Directory Path Selection

This is a Dialog which shows the Directory Structure to select the image database path. Following are the important function to create this dialog.

1. To get the Drives from the Operating System

   GetLogicalDriveStrings(Bufsize,Buf);

2. To Get the Drive Type of the Dirve.

```
GetDriveType ((LPCTSTR) tmp);
```

3. To Get the File and Directory list from a specific Path we need a Handle to search and a structure in which we will get the information about files and two functions described below

```
HANDLE hFind;
WIN32_FIND_DATA fd;
FindFirstFile ((LPCTSTR) Usepath, &fd); // To Find a File
FindNextFile (hFind, &fd) // To find Next File
```

4.2.6 Wait While Performing Operation Dialog Box

A Wait while Performing Operation Dialog Box displayed when user perform any operation along with the appropriate message.

```
CWaitWhilePerformingOperation* WaitWindow;
WaitWindow = new CWaitWhilePerformingOperation();
WaitWindow->Create(IDD_WAITWHILEPERFORMOPERATION, this);
WaitWindow->ShowWindow(SW_SHOWNOACTIVATE);
WaitWindow->m_Text.SetWindowText("Message");
//Operation Code
WaitWindow->DestroyWindow();
delete WaitWindow;
```

4.2.7 Accelerator Keys

Accelerator Keys are defined so that user or operator can perform functions quickly and without using mouse and if the mouse is not working then these keys help operator to use the software with the help of Accelerator Keys. And we have overloaded the function *PreTranslateMessage(...)*

```
BOOL CFaceRecDlg::PreTranslateMessage(MSG* pMsg)
{
        if (pMsg->wParam == 27)
                return TRUE;
        if(pMsg->message == WM_SYSKEYDOWN && pMsg->wParam > 32)
        {
                switch ((char) pMsg->wParam)
                {
                        case 'D':
                                m_DBImagePath.SetFocus();
                                m_DBImagePath.SetSel(MAKEWORD(-1,0));
                                return TRUE;
                        .       .       .       .
                        .       .       .       .
                        .       .       .       .
                        case 'E':
                                m_DisEigenface.SetCheck(!m_DisEigenface.GetCheck());
                                m_DisEigenface.SetFocus();
                                return TRUE;
                        case 'P':
                                OnKlt();
                                return TRUE;
                        case 'G':
                                m_RecImagePath.SetFocus();
                                m_RecImagePath.SetSel(MAKEWORD(-1,0));
                                return TRUE;
                        .       .       .       .
                        .       .       .       .
                        .       .       .       .
                        case 'X':
                                OnCancel();
```

```
                return TRUE;
            default:
                break;
        }
    }
}
```

4.2.8 Proper Messages

Proper Error, Information and Question Messages are shown at proper time along with the ICONS so at first sight user can guess which kind of message is displayed by the software. Sample Messages are

```
AfxMessageBox("Matlab Server not initialized.",MB_ICONERROR);
AfxMessageBox("KL-Transform(PCA) successfully
                    completed.",MB_ICONINFORMATION);
AfxMessageBox("Unsuccessful recognizing due to Invalid command to load
                    file.",MB_ICONERROR);
AfxMessageBox("Are you sure you want to exit Face Recognition
                program?",MB_ICONQUESTION|MB_YESNO)
```

4.2.9 Help File

A Help file is being provided to the one who buy our software so that if he feel difficulty in using the software can consult the help file. There is a button of Help and also User can call the Help File by *Pressing F1. OnHelpInfo()* Function is overwrite to invoke help on F1.

```
OnHelpInfo(HELPINFO* pHelpInfo)  //Function to overwrite for help
{
    OnHelpinformation();
    return false;
}
void CFaceRecDlg::OnHelpinformation()
```

{WinHelp(0,HELP_FINDER); }

4.3 Algorithm and Pseudo Code

4.3.1 Algorithm for KL-Transform

Begin

Read M

I_n = image n of training set for n =1........M

A = Mean of all images of training set;

$X_n = I_n - A$ Mean Deviation Image for n=1 ... ... M

Normalize X_n

Let $L=X^T X$

Calculate $LV_n = \lambda_n V_n$ for n =1... ... ... M

Sort V's according to λ's (in Descending Order)

Calculate eigenfaces Un = X x V_n

Normalize $U_n = V_n / \| V_n \|$

Store values A, V and U.

End

4.3.2 Pseudo Code for KL-Transform

//To load data store in a file whose name is send as a parameter

load("Filename.mat");

//To save data in a file whose name is send as a parameter and also the variables which we want to store

save("Filename", Variable name);

// To read image and it takes two parameter. One is the image path including extension and other is the image type basically extension.

imread("Image File Path",'Image Type');

// To Calculate Mean Image there is no formulae available according to our algorithm demands so it is done manually

```
A=A+Images(:,n);
A=A./M;
```

// Calcuating Mean Deviation and Normalizing it

```
Images(:,n)=Images(:,n)-A;
Images(:,n)=Images(:,n)./norm(Images(:,n));
```

//Calculating Covariance Matrix

```
L=Images'*Images;
```

//Calcuating Eigen Values

```
[V lambda]=eig(L);
```

// Calculating U_n and Normalizing it.

```
U(:,n)=Images*V(:,n);
U(:,n)=U(:,n)./norm(U(:,n));
```

4.3.3 Algorithm for Recognition

Begin

I = Test Image

$y = U^T (I - A)$

Let err = sqrt $(||I\text{-}A||^2 - ||y'y||)$

If err < definedThreshold1 **then**

Not a Face

else

Pass it through Enhancement Algorithm;

Project the Test Image onto Eigenspace;

If error between test image and any Training Set image<threshold2

```
                    Matched With Database
            else
                    Pass it through Glasses Enhancement Algorithm
                    If error between test image and any Training Set then
                                        image<threshold2
                        Matched With Database
                    else
                        Not in Database
                    end if
            endif
        endif
    End
```

4.3.4 Pseudo Code for Recognition

// To see whether the input image is face or not

```
err=sqrt(norm(TestImage)^2-Y'*Y)/10000;
```

//Enhancement of Image if needed and it takes four parameter TestImage, its width, whether or not to display Enhancement Area, Whether or not to display Enhanced Image.

```
Enhancement(TestImage, Width, DisEnhancementArea, DisEnhancedImage);
```

// If not matched check for Glasses Enhancement and it takes three parameter TestImage, its width, Whether or not to display Enhanced Image.

```
GEnhancement(Test Image, Width, DisEnhancedImage);
```

//Match with Database and it take four paraments: mfiles path, value of U, value of A and value of Y.

```
matchwithdatabase(mfilespath, U, A, Y);
```

4.3.5 Algorithm for Cold Effect Enhancement

```
Begin
        Get image, dimension and display options;
        Get the points to help enhance the area;
        Check image need enhancement or not;
        If Image need Enhancement then
                Select the area to enhance;
                Enhance the Area;
                Put it back in original Image;
                Display the enhancement area and enhanced image based on
                display options;
                Return the enhanced image back to recognition process;
        Else
                Return the original test image to recognition process;
        End
End
```

4.3.6 Pseudo Code for Cold Effect Enhancement

```
// Get specific points to be used in enhancement and checking values
        mid=int16(m/2);
        one4rth=int32(m/4);

//Get Values less than 100
        if(ImagetoEnhanse(i,j)<100)
                GetVals=GetVals+1;
        End

//Check whether image need enhancement or not
        if(GetVals>=thresholdtoselect)
```

```
//Select Area for enhancement
    for i=one4rth-FivePercentofm:m-FivePercentofm
        if(i>mid & count>3)
            if(i<mid+one4rth)
                .   .   .
                .   .   .
        else
                .   .   .
                .   .   .
        end
        count=1;
      else
        count=count+1;
      end

//Enhance Image
    if(ImagetoEnhanse(i,j)<60 & ImagetoEnhanse(i,j)>45)
            .   .   .
            .   .   .
        elseif (ImagetoEnhanse(i,j)<100 & ImagetoEnhanse(i,j)>60)
            .   .   .
            .   .   .
    end

//Display Enhancement area and enhanced image based on selection
    if(DisEnhaceArea=='y')
            .   .   .
            .   .   .
    end
    if(DisEnhImage=='y')
            .   .   .
            .   .   .
```

```
        end

// Return Image
        return image;
```

4.3.9 Algorithm for Sunglasses Filtering

Begin

Get image, dimension and display options;

Get the points to help filter the sunglasses;

Check image need filtering or not;

If Image need sunglasses filtering **then**

Filter the image and remove the sunglasses at its best;

Display Filtered image based on display options;

Return the Filtered image to recognition process;

Else

Return input image as it is to recognition process;

End

End

4.3.10 Pseudo Code for Sunglasses Filtering

```
// Get specific points to be used in Filtering and checking values
        fourpercent=int32(m*.04);
        eightpercent=int32(m*.08);

//Get Values less than 45
        if(ImagetoEnhanse(i,j)<45)
                GetVals=GetVals+1;
        End
//Check whether image need Filtering or not
        if(GetVals>=thresholdtoselect)
```

```
//Filtering Image and replacing the sunglasses pixel with nose tip values
        if(ImagetoEnhanse(i,j)<40)
            .    .    .
            .    .    .
         end

//Display Enhancement area and enhanced image based on selection
    if(DisEnhImage=='y')
        .    .    .
        .    .    .
    end

// Return Image
    return image;
```

4.3.9 Algorithm for Reconstruction

```
    Begin
        Get name of image to be reconstructed;
        Normalize image and subtract Mean
        Reconstruction RI=U*Y+A;
        Display Reconstructed Image;
    End
```

4.3.10 Pseudo Code for Reconstruction

```
// Load Training Set Information
    load("Training set information file path");

//Load Mean Image
    load("Path to Mean Image file");
```

```
//Load Values of U
        load("Path to Mean Image file");

//Reconstruct Image
        U*Y+A;

//Display Reconstructed Image
        imagesc(ReconstructedImage);
```

Chapter 5

Testing and Results

5. TESTING AND RESULTS

The overall objective of the testing process is to identify the maximum number of errors in the code with a minimum amount of efforts. Finding an error is thus considered a success rather than failure. On finding an error, efforts are made to correct it.

5.1 Testing The Software

Our Software is basically divided into three components or modules (KL-Transform, Recognition and Reconstruction). Initially unit test was performed on every unit. Syntax errors were removed and the validation checks were tested and corrected entirely. For semantic errors, every program unit was tested with the help of test data.

After this all three modules are combined in one complete software as the integration completed. Again the Syntax and semantic errors were checked and removed. After the completion of individual testing of all the modules, the modules were integrated and testing phases were applied, then errors were checked and removed.

In the next step, the system test was applied and all the errors were checked and removed. When all these errors were detected and removed, the final test was applied to the system, to check whether these changes are affecting the remaining parts of the program unit or not. The errors occurred were removed. After this checking, all the functionality was checked and found correct.

To Test our software the two test sets and training sets have been chosen to verify that our software is working good or not and named:

- Test Set 1 and Traning Set 1.
- Test Set 2 and Training Set 2.

5.1.1 Test Set 1 and Training Set 1

Training Set consist of 4 infrared images of four different persons in order to verify that our software works as efficiently on smaller training set as it works on larger training set.

Test Set Contains 11 images of different persons who are in database and are not in database and 1 out of 11 wore the sunglasses. One image is not a face and the dimension of other image is changed in order to verify the program. So the test set contains the total of 13 images.

This set also verifies if the number of training set images is less than the test set image, this algo doesn't lead us to False Acceptance or False Rejection.

5.1.2 Test Set 2 and Training Set 2

Training Set consist of 72 infrared images of 8 different persons means 9 different expressions for each person in order to verify that our software works efficiently on larger training set.

Test Set Contains 20 images of different persons who are in database and are not in database where two out of 20 wore sunglases and there are some images out of 20 who are affected by the temperature. One image is not a face and the dimension of one other image is changed in order to verify the software and one image is self made image so to verify that software works when a picture came who does not have the features of face but only have the Shape. Thus making grand total of 23 images. Expressions other than used in the traning set are used in this test set for the persons who are from traning set.

This set also verifies if the number of training set images is greater than the test set image, this algo doesn't lead us to False Acceptance or False Rejection.

5.2 Results

Following are the results obtained after applying the test set to the software "Face Recognition through Infrared Images and Eigenfaces".

5.2.1 Results of Test Set 1 and Training Set 1

Following results are obtained when Test Set 1 and Training Set 1 is used or applied to software.

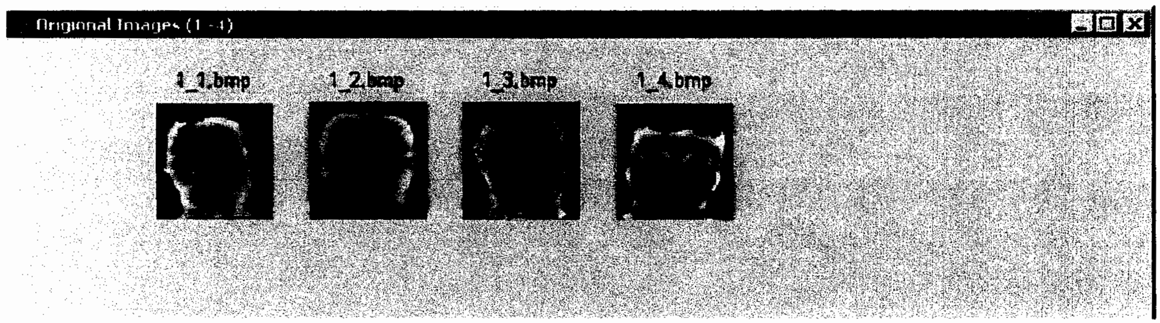


Figure 5.1 Sample Images of Traning Set 1

Figure 5.1 shows the sample images of the traning set 1. and different expressions used for different person can be seen.

Figure 5.2 Mean Image of Traning Set 1

Figure 5.2 shows the Mean imageof the traning set 1.

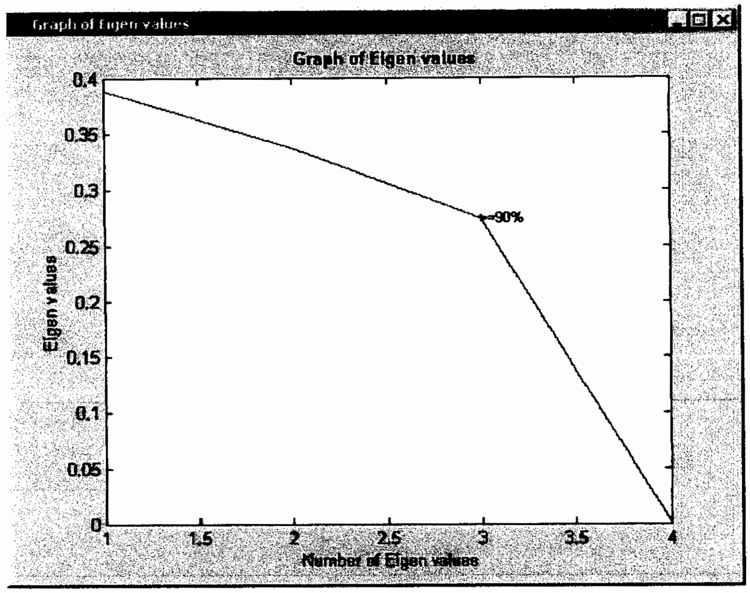


Figure 5.3 Graph of Eigenvalues Vs Number of Eigenvalues

Figure 5.3 shows the Graph of Eigenvalues Vs Number of Eigenvalues and a mark on 90% Eigenvalues which are chosen to define our Eigenspace. Figures 5.4 shows the eigenfaces formed depending on the eigenvalues.

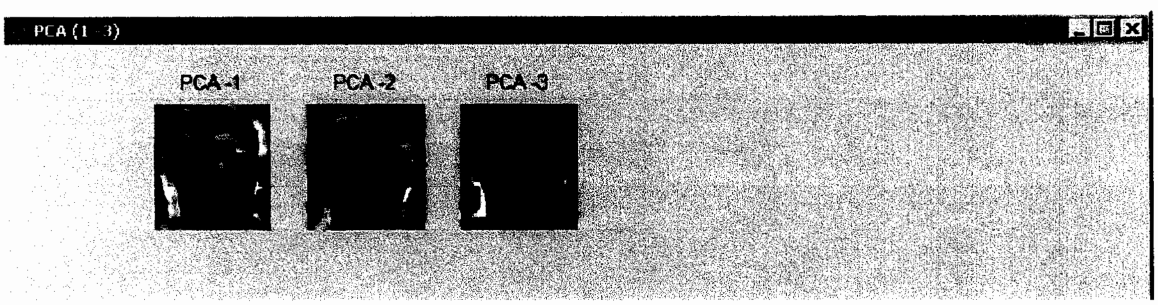


Figure 5.4 Eigenfaces

When the image which is not a face is sent, our algorithm applies some mathematics and check the distance of the image from the eigenspace. If the distance is greater than the defined threshold, the software immediately Display the image with title "Not a face" as shown in Figure 5.5.

Figure 5.5 Not a face

When a face which does not belong to authorized person is sent, our algorithm calculates the distance of image from the eigenspace. If the distance is greater than the defined threshold then the software display the image with title "Not in database" as shown in Figure 5.6.

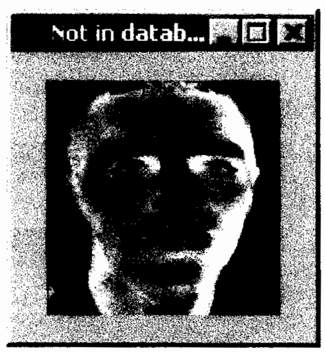
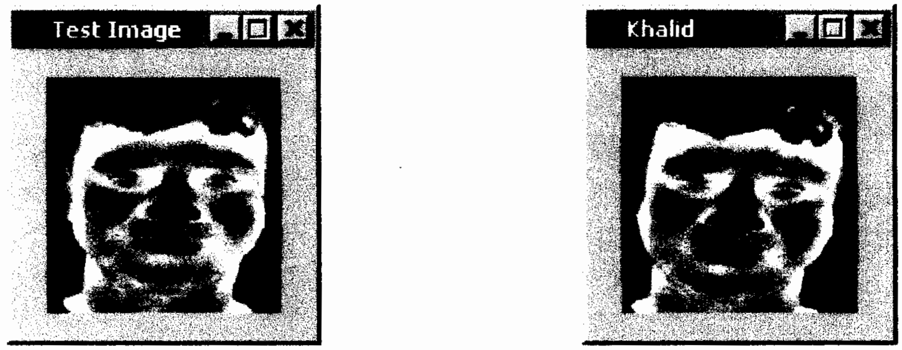


Figure 5.6 Not In Database

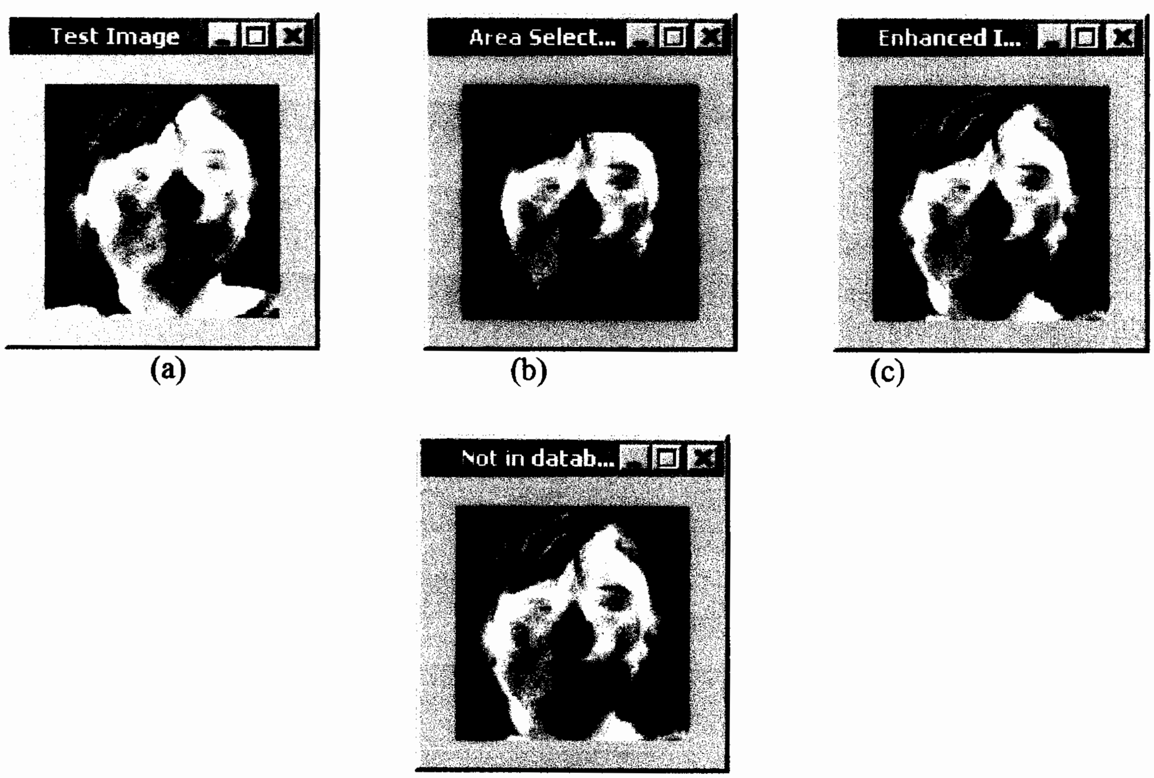
When image is of the person that belongs to training set but having different expression and not affected by cold, our algorithm project the image on to the eigenspace and. If the person is from the training set then it display the matched image as shown in Figure 5.7.



(a) (b)

Figure 5.7 Recognition of Face

When image is of the person that belongs to training set affected by the temperature i.e. the person came from the cold, Our algorithm checks the nose and checks pixel value and if found affected then it enhances the image and then project it on to the eigenspace. If the image is not of a person belongs to training set then display the image with title "Not in database" as shown in Figure 5.8.

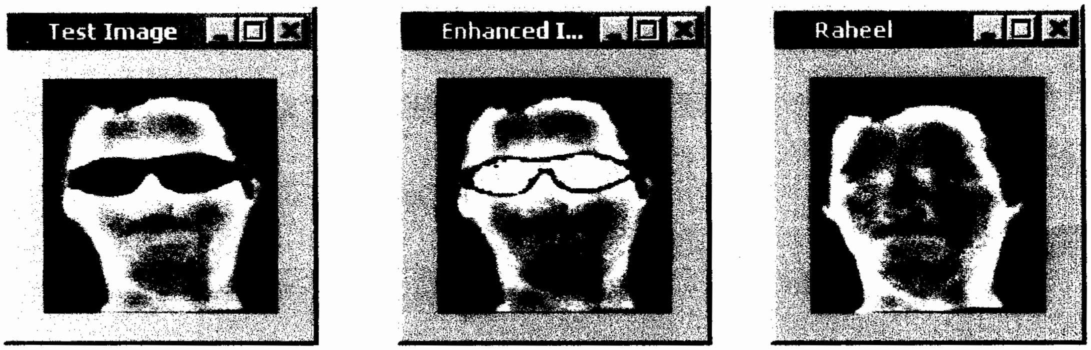


(a) (b) (c)

(d)

Figure 5.8 Recognition after passing through Enhancement Algorithm

When image is of the person that belongs to training set but wearing sunglasses, Our algorithm removes the glasses and replace the glasses value with the nose tip pixel value and then project it to the eigenspace and if found a match then display the matched image as shown in Figure 5.9.



(a) (b) (c)

Figure 5.9 Recognition after passing through Sunglasses Filtering Algorithm

5.2.2 Results of Test Set 2 and Training Set 2

Following results are obtained from Test Set 2 and Training Set 2.

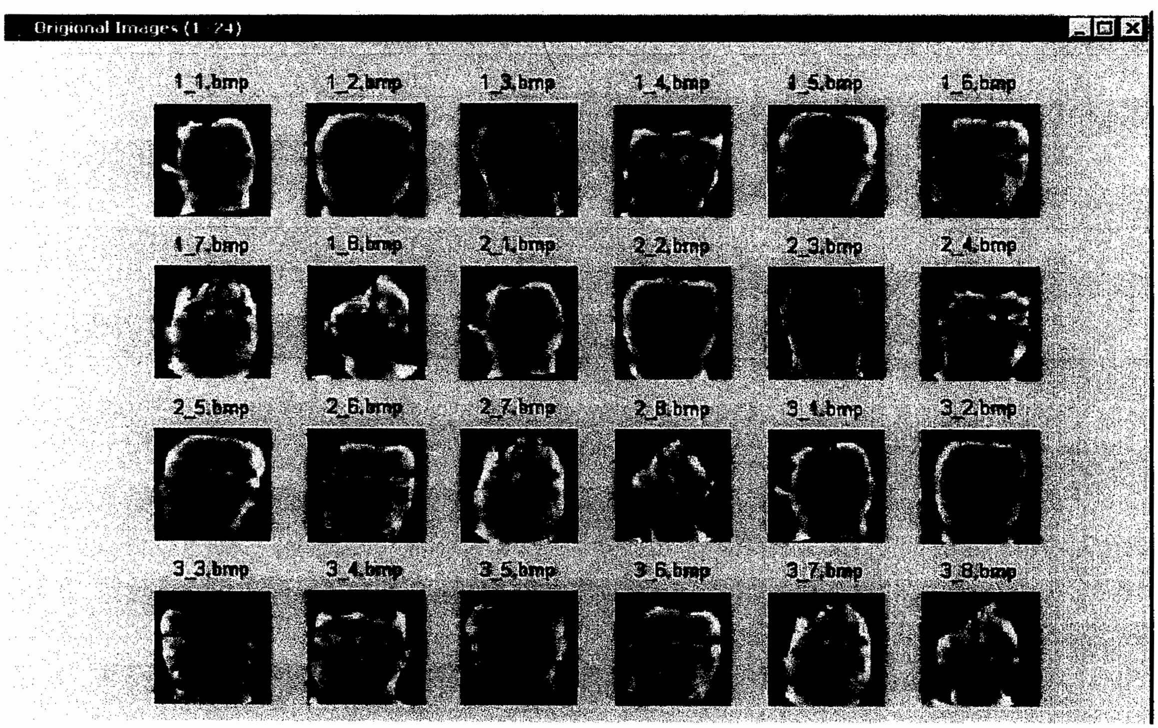


Figure 5.10 Sample Images of Traning Set 2

Figure 5.10 shows the sample images of the traning set 2. and different expressions used for different person can be seen. Figure 5.11 shows the Mean image of the traning set 2.

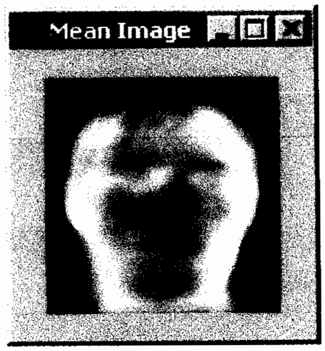


Figure 5.11 Mean Image of Traning Set 2

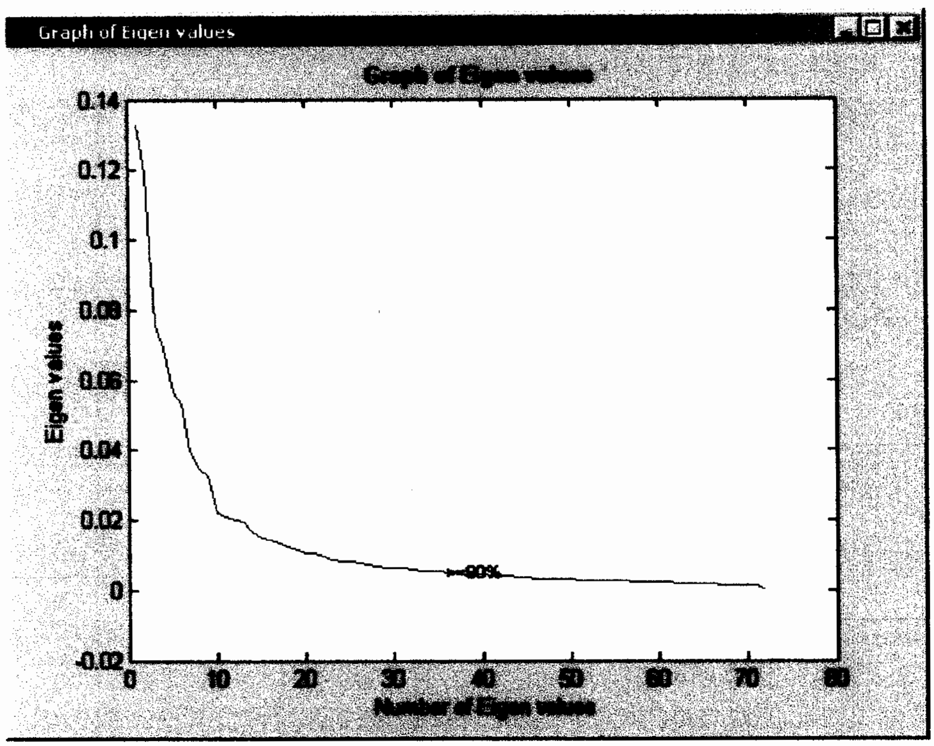


Figure 5.12 Graph of Eigenvalues Vs Number of Eigenvalues

Figure 5.12 shows the Graph of Eigenvalues Vs Number of Eigenvalues and a mark on 90% Eigenvalues which are chosen to define our Eigenspace.

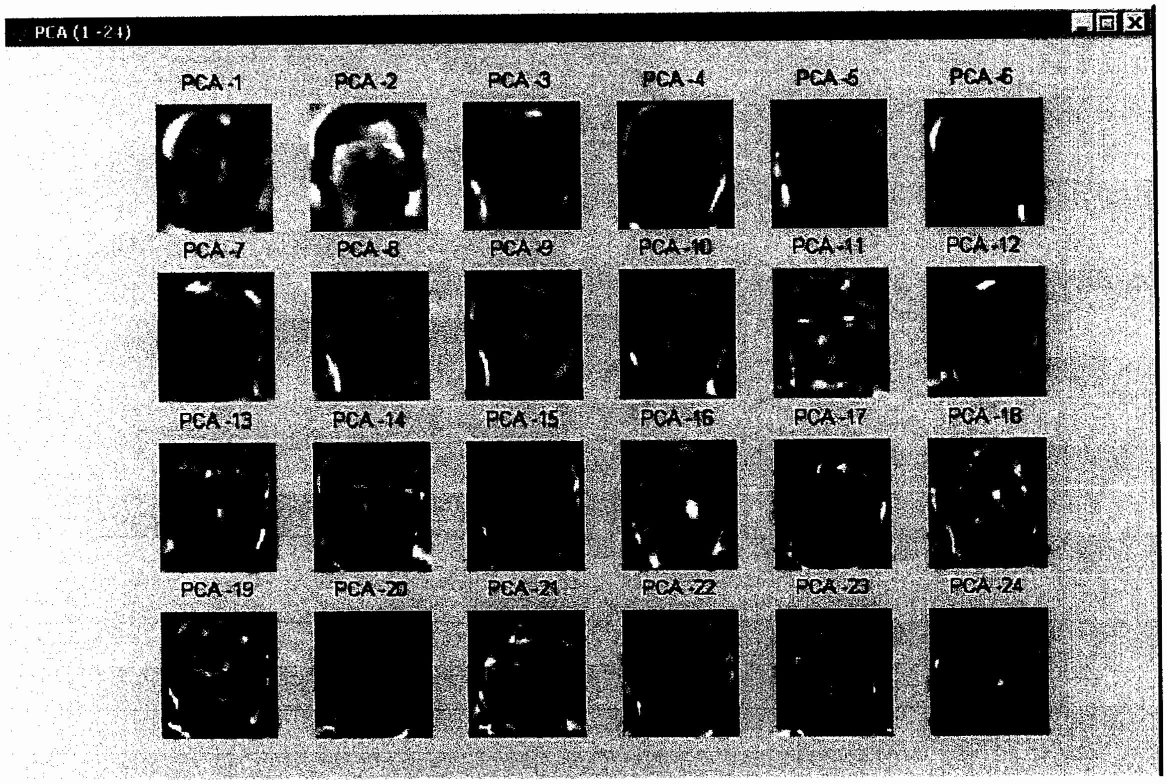


Figure 5.13 Eigenfaces

Figures 5.13 shows the eigenfaces formed depending on the eigenvalues.

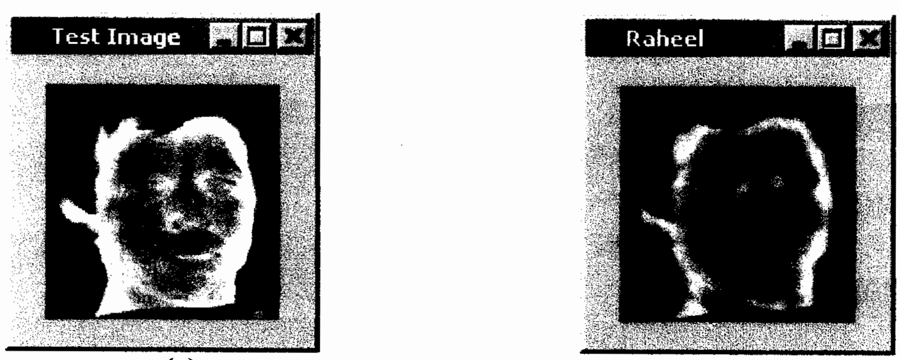
When the image which is not a face is sent, our algorithm applies some mathematics and check the distance of the image from the eigenspace. If the distance is greater than the defined threshold, the software immediately Display the image with title "Not a face" as shown in Figure 5.14.

Figure 5.14 Not a face

When a face which does not belong to authorized person is sent, our algorithm calculates the distance of image from the eigenspace. If the distance is greater than the defined threshold then the software display the image with title "Not in database" as shown in Figure 5.15.

Figure 5.15 Not In Database

When image is of the person that belongs to training set but having different expression and not affected by cold, our algorithm project the image on to the eigenspace and. If the person is from the training set then it display the matched image as shown in Figure 5.16.



(a) (b)

Figure 6.16 Recognition of Face

When image is of the person that belongs to training set affected by the temperature i.e. the person came from the cold, Our algorithm checks the nose and checks pixel value and if found affected then it enhances the image and then project it on to the eigenspace. If the image is not of a person belongs to training set then display the image with title "Not in database" as shown in Figure 6.17.

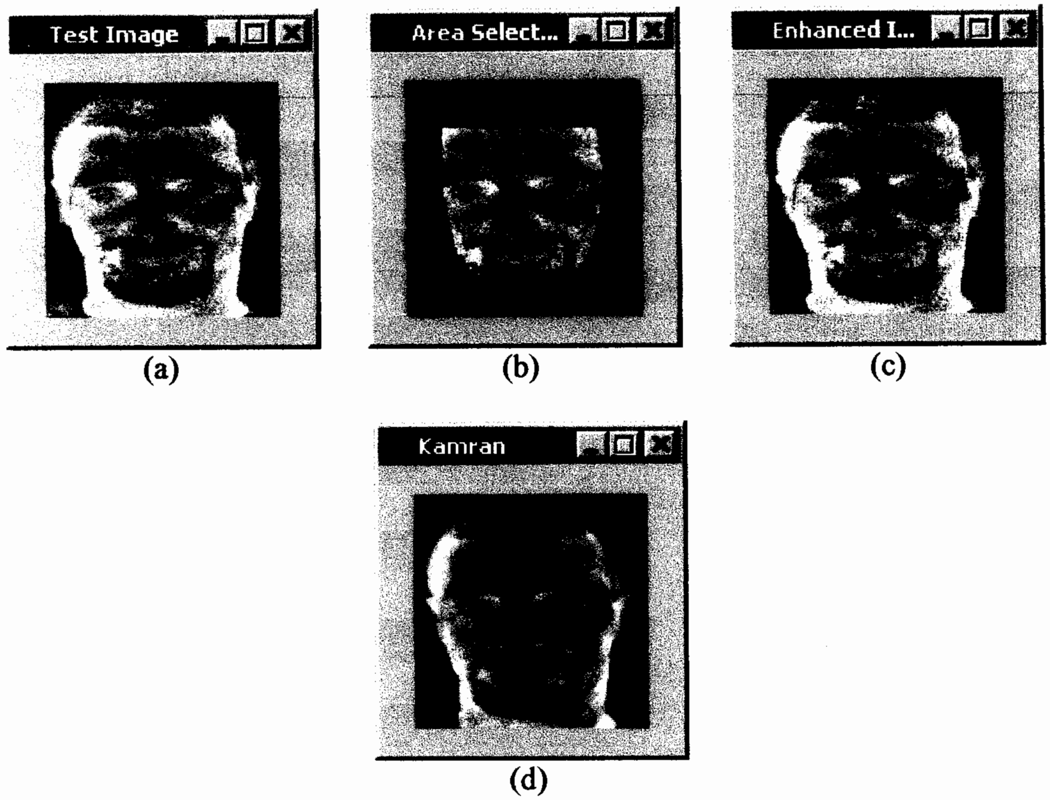


Figure 5.17 Recognition after passing through Enhancement Algorithm

When image is of the person that belongs to training set but wearing sunglasses, Our algorithm removes the glasses and replace the glasses value with the nose tip pixel value and then project it to the eigenspace and if found a match then display the matched image as shown in Figure 5.18.

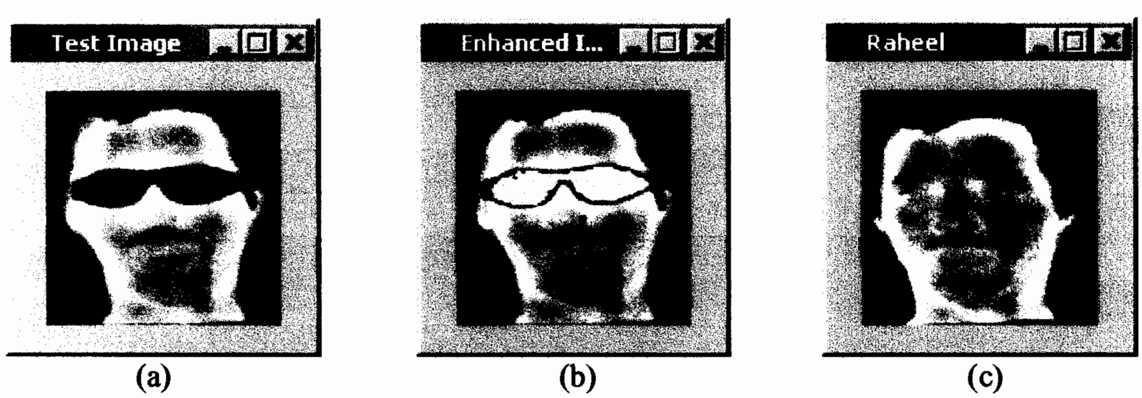


Figure 5.18 Recognition after passing through Sunglasses Filtering Algorithm

If we recall our scope of project, we have to increase the accuracy, Handle different expressions, Facial hair, Temperature effect and sunglasses filtering. Above results prove all the things. Different expressions handled by the software, facial hair are handled using threshold, Temperature effect algorithm is working well, and Sunglasses algorithm is also working quiet efficiently. Since all the things are working well hence the accuracy automatically increases. The Testing proves this software to be efficient, accurate, fast and reliable software for Recognition of infrared face images using the technique Principal Component Analysis. This software is up to the mark to be called as one of the good software which can compete at the university level and at the market level of Pakistan.

5.3 Conclusion

This project is a research project and we have extended the work done by the Rose Cutler in 1996 on Infrared Face Recognition using Eigenface technique. We have introduced in it two more algorithms for temperature control and sunglasses filtering in order to increase its efficiency from 96.6% to as much as we can increase and the testing gives us the results up to our expectations.

We have shown that the eigenface technique for face recognition successfully applies to infrared images. And also seen result using different expressions of face and sees the result and found them very satisfactory. We have also handled the temperature effect now if someone came from the cold will surely be recognized and that will automatically decrease the False Rejection Rate. The sunglasses algorithm is also implemented and we get satisfactory result from that algorithm and this also has reduced the False Rejection Rate.

The one and only danger while developing this algorithm is that it might lead us to False Acceptance but after testing and testing on small training set and large training set it has been proved that it has not increased the false Acceptance Rate. This way this software has become one of the best Face Recognition software that can compete at University and Market level of Pakistan software Industry.

There are many future experiments that could be done to extend this study. Clearly a much larger training and test set would be interesting. Also, higher resolution infrared cameras are available, and should yield greater subsurface face detail. Finally, utilizing the eigenface technique to find the eyes, nose, and mouth would be most useful in automating the alignment procedure.

This algorithm is not limited to the face recognition or Biometrics but can be used in Character Recognition, Weather Forecasting, Missiles, Dimension Reduction, Satellite Images, compression and ca be used in the encryption of Data. And one of the best aspects of the algorithm is that it is fast.

In future, vast fields are open for this field and it can be used on such places where u need speedy results and speedy work done.

Bibliography and References

Bibliography and References

Research Papers

[1] Chellapa, R., et. Al., "Human and Machine Recognition of Faces: A Survey", UMCP CS-TR- 3339.

[2] K. Fukunage, "Introduction to Statistical Patern Recognition" (2^{nd} edition), Academic Press, New York, 1990.

[3] Moghaddam and Pentland, "Face Recognition using View-Based and Modular Eigenspaces, Automatic Systems for the Identifications and Inspection of Humans", SPIE Vol. 2277

[4] M. Kirby and L. Sirovich, "Application of the Karhunen-Loeve procedure for the characterization of human faces". IEEE Transaction on pattern Analysis and Machine Intelligence. Vol 12. no. 1. pp. 103-108, January 1990.

[5] M. Turk, A. Pentland, "Face Recognition using Eigenfaces", IEEE Conf. on Computer Vision and Pattern Recognition, 1991

[6] Pentland, et. Al. "Experiment with Eigenfaces", MIT VISMOD TR-194

[7] Rose Cutler, "Face Recognition Using Infrared Images and Eigenfaces", 1996.

[8] Steve, Peter and Ingemar, "Face Recognition using Mixture-Distance and Raw Images", IEEE International Conference on Systems, 1997.

[9] Turk, M., and Pentland, A., Eigenfaces for Recognition 1991, Journal of Cognitive Neuroscience, Vol. 3, No. 1, pp 71-86.

[10] T. Fromhez, "Face Recognition: A summary of 1995-1997". Technical Report ICSI TR-98-027, International Computer Science Institute, Berkeley, 1998.

[11] U. Grenander. A. Srivastava and C Hesher, 01] "Models for Statistical Analysis of Range" Images, November 2001.

Books

[12] Digital Image Processing. Rafael C. Gonzalez and Richard E. Woods. 1993 Prentice Hall Press.

[13] Digital Image Processing Jan Teuber, 1992 Prentice Hall Press.

[14] Linear Algebra and its Applications, David C Lay (University of Maryland). 1994 Addison Wesley Publishing Company.

[15] Numerical Recipies in C. Press, W. H. 1994 Cambridge University Press.

[16] Software Engineering A Practitioner's Approach (Fourth Edition) by Roger S. Pressman 1992, McGraw-Hill Companies Inc.

Others

[17] Biometric Security System Technology Page.

[18] MATLAB (Online Documentation).

[19] Principal Component Analysis An Introduction, Thomas B. Moeslund. Laboratory of Computer Vision and Media Technology Aalborg University, Demark.

[20] US Patent 5,163,094, Technology Recognition Systems

Appendix A

User Manual

Overview

This is a program developed for security purpose and can be used in buildings, offices and sensitive areas in order to avoid the unauthorized person's access to particular area. This program uses an easy to use interface and a very accurate, fast and reliable technique called **Principal Component Analysis** to recognize the infrared face image of a person to grant permission to access or enter a particular area.

This software is developed in Visual C++ and Matlab. This software uses the computational efficiency of matlab and rests of the things are handled in Visual C++.

This Program is developed by: Software Development Team

To Get this software Contact: Contact Me

To learn about the program see the Main Window (Application)

Main Window (Application)

This is what the main program window looks like:

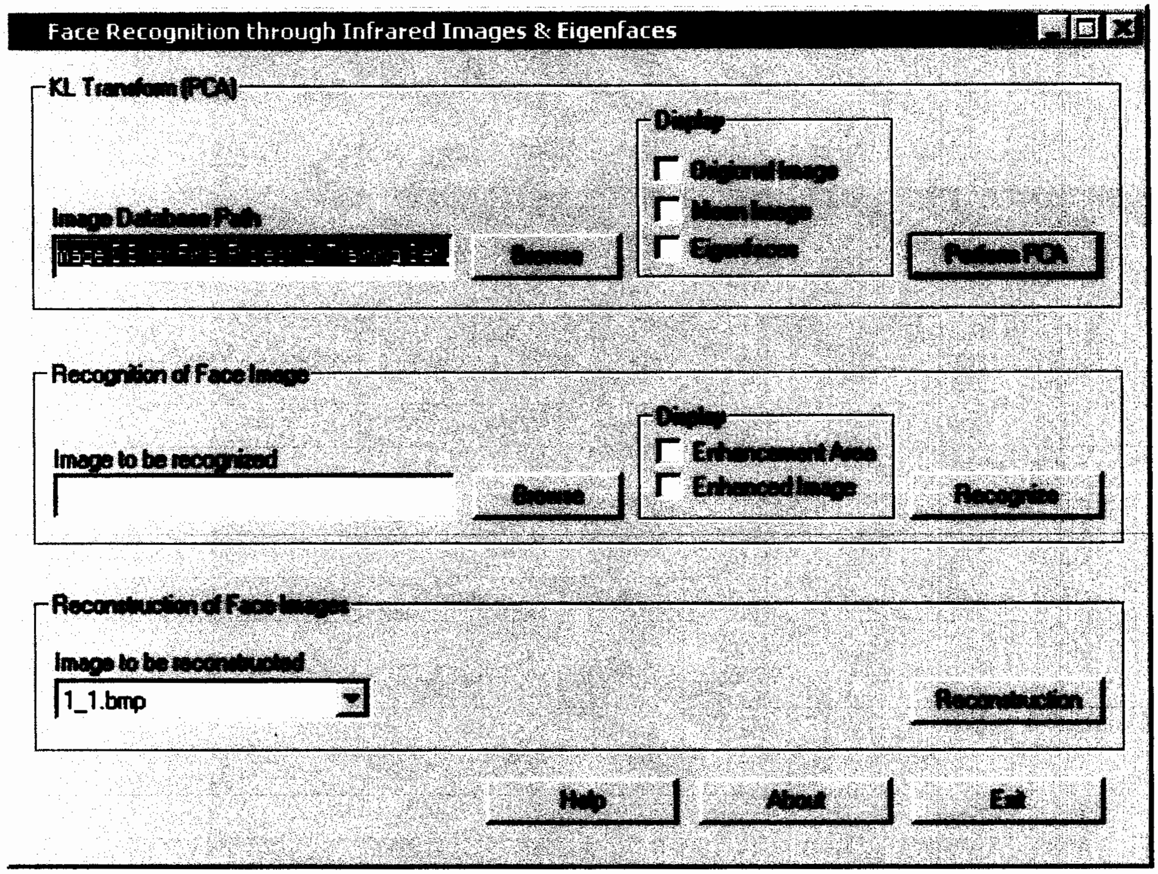


Figure 1. Main Window

KL-Transform (PCA)

Image Database Path: It shows the path to the image database folder and user can type the path directly here if he knows the exact path. The focus can be moved to the filed using accelerator key **ALT+D**.

Browse : See Image Database Browse Dialog

Display: Display option helps user select which images should it show when he run KL-Transform.

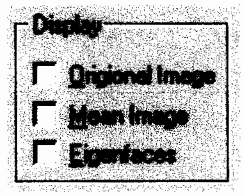


Figure 2. Display Options

Original Image: If user wants to see the original images then he should select this option. The accelerator key is **ALT+O**.

Mean Image: If user wants to see the Mean Image then he should select this option. The accelerator key is **ALT+M**.

Eigenfaces: If user wants to see the Eigenfaces then he should select this option. The accelerator key is **ALT+E**.

Perform PCA: To perform the PCA or KL-Transform operation press this button and it will display <u>Training Set Other Information</u> to enter the information related to Image, Training Set, and Subject Names. and after saving the information it perform the PCA or KL-Transform and display the images (<u>See Sample Images</u>)depending on the choice of the user. and during this whole procedure shows the wait window

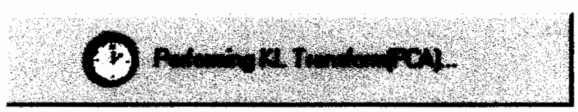


Figure 3. Wait Window for KL-Transform

It always shows the <u>Graph of Eigenvalues</u> which tells us about the no of Principal Component which we have selected. The accelerator key to perform this operation is **ALT+P**.

Recognition of Face Image

Image to be Recognized: It shows the path to the image which is to be recognized by this software user can type the path directly here if he knows the exact path. The focus can be moved to the filed using accelerator key **ALT+G**.

Browse: See <u>Image to be Recognized Browse Dialog</u>

Display: Display allow user to select whether or not he want to see the Enhancement Area** of image enhancement and the enhanced image if image need enhancement. The Test Image is displayed and the matched image is also displayed (if match is find) always.

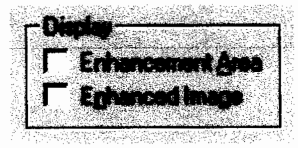


Figure 4. Display Options

Enhancement Area: If user wants to see Enhancement Area then he should select this option. The accelerator key is **ALT+A**.

Enhanced Image: If user wants to see the Enhanced Image then he should select this option. The accelerator key is **ALT+N**.

Recognize: To Recognize the test image Press Recognize button. If the image is not a face then it will only display the test image with title "Not a Face" and if it is a face and the person is not found in database then display the test image showing title "Not in Database" and if found in database then display the test image, enhancement area, enhanced image and the reconstructed image of the Person showing the title "His Name". During Recognition process user see the wait window.

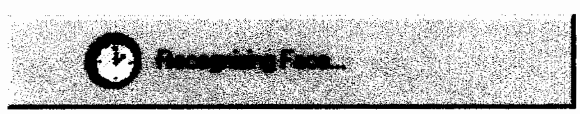


Figure 5. Wait Window for Recognition

The accelerator key to perform this operation is **ALT+R**. Some sample results of recognition process are <u>Recognition Process Results</u> .

Reconstruction of Face Image

Image List to be Reconstruct: This is the list of the Original Images names. and user can select which image he wants to reconstruct. The reconstructed image will be little different and blur because we have reduced the dimension of the eigenspace.

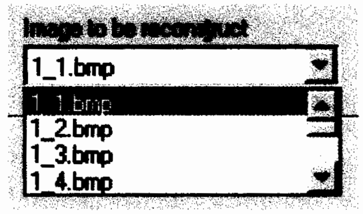


Figure 6. Image Name List

The accelerator key to select image from the list is **ALT+T**.

Reconstruction: To Reconstruct Original Image from available PCA press reconstruction button. The Image will be reconstructed but a little different and blur because of the reduced dimension eigenspace. and during the reconstruction of face image user see the wait window

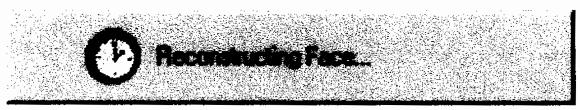


Figure 7. Wait Window for Reconstruction

The accelerator key to perform reconstruction is **ALT+C**. Sample result of Reconstruction process is Reconstruction Process Result

** Enhancement Area is only shown when the temperature affected face is enhanced; it is not shown when the glasses filtering is done.

Image Database Browse Dialog

This is what the Image Database Browse Dialog looks like and it is popup in front of the Main Window

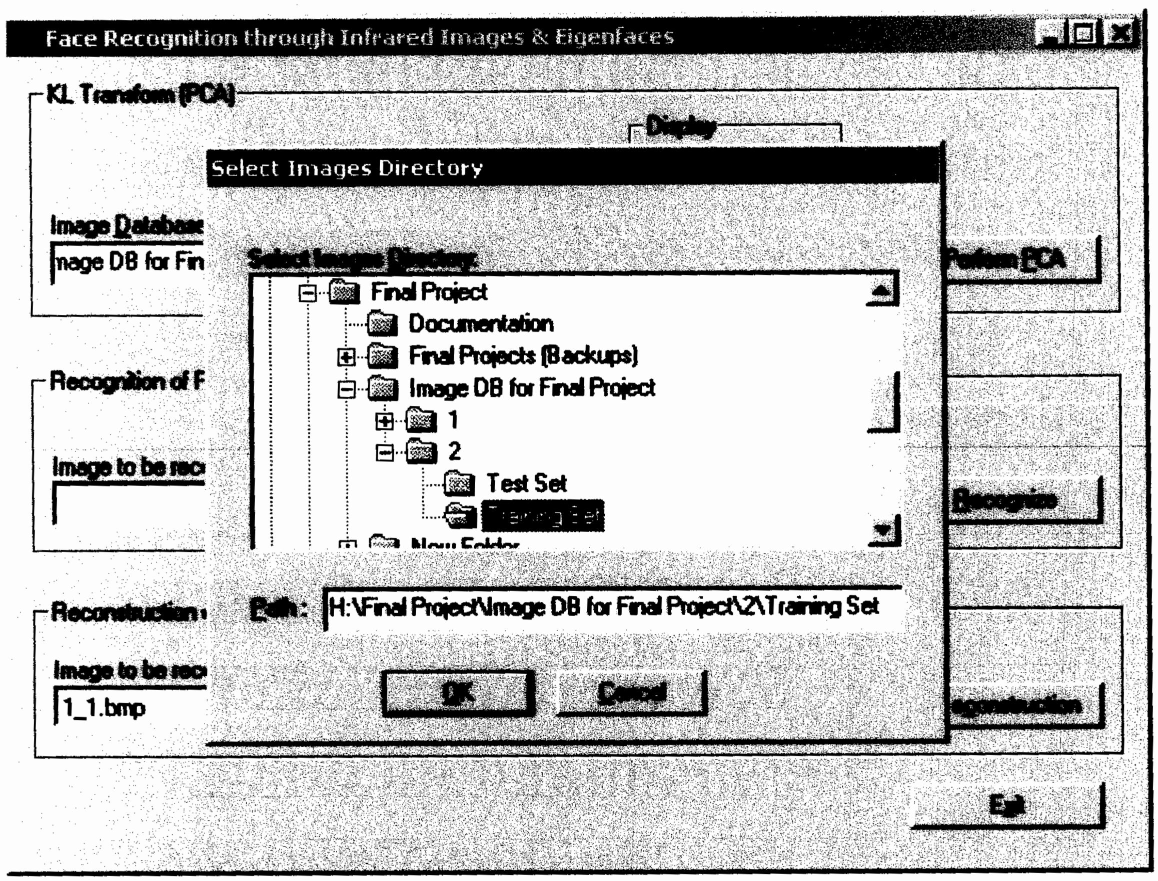


Figure 8. Select Images Directory

Select Image Directory: it is a directory control and user can select the image database path through this control and the accelerator key to move focus to directory control is **ALT+D**.

Path: It shows the path selected from the directory control and if user knows the exact path then he can type it directly here. The accelerator key to select the contents of this field is **ALT+P**.

Ok: After selecting the directory the user can press ok to move back to main window and the Image Database Path field will show the path selected by user. Te accelerator key to press OK button is **ALT+O**.

Cancel: If User thinks that the previous path was the correct path and I do not want to change it then he can press Cancel to return to main window without any changes to Image Database Path. The accelerator key to press Cancel button is **ALT+C**.

Training Set Other Information Dialog Box

This is what the Image Database Browse Dialog looks like and it is popup in front of the Main Window

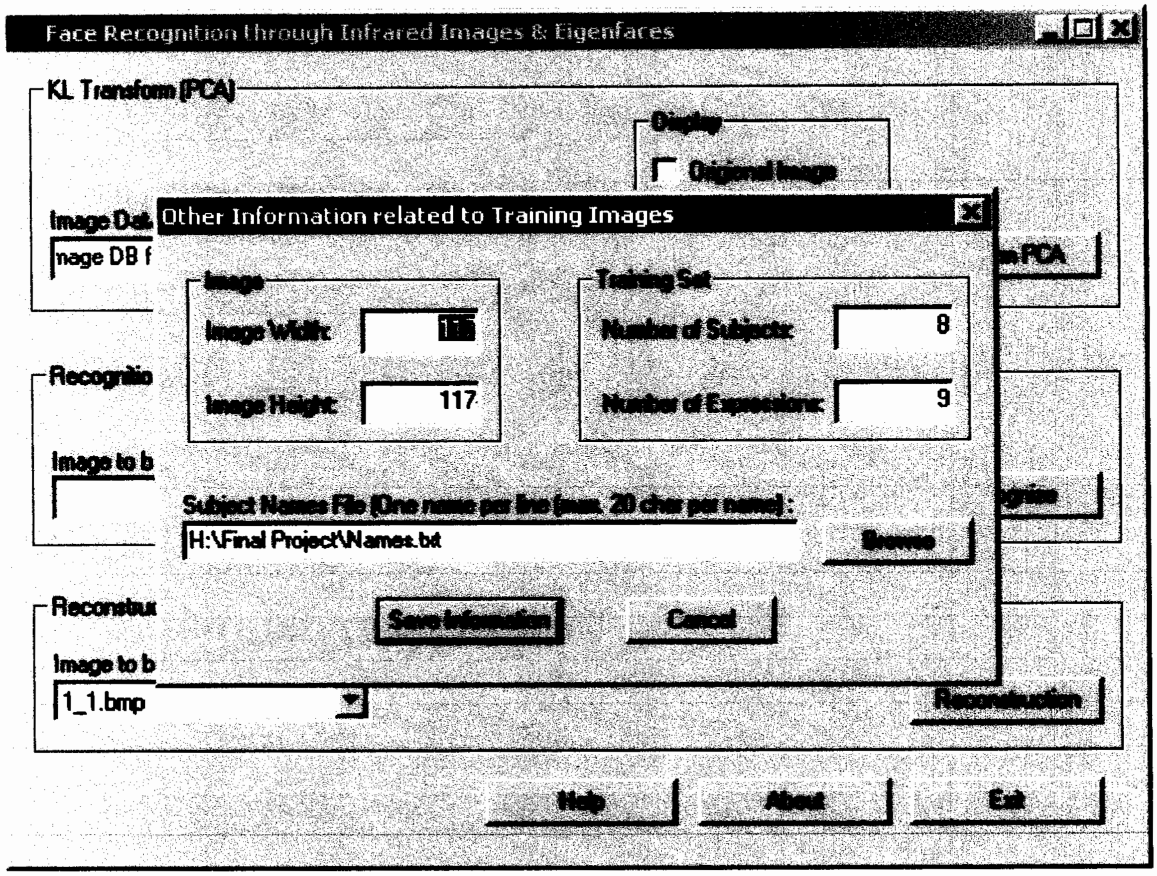


Figure 9. Other Information Dialog Box

Image**: This option is used to enter the image dimension information.

Image Width: User specifies the Image Width in this column. the accelerator key to move focus to this field is **ALT+W**.

Image Height: User specifies the Image Height in this column. the accelerator key to move focus to this field is **ALT+H**.

Training Set: This option is used to enter the training set or image database information.

No of Subjects: User specify the number of subjects used to form training set in this column. The accelerator key to move focus to this field is **ALT+U**.

No of Expressions: User specify the number of expressions for each subject in this column. The accelerator key to move focus to this field is **ALT+E**.

Subject File Name: This field is used to specify the names of the subjects used in the training set. if user knows the exact path he can enter the path here. The accelerator key to move focus to this field is **ALT+N**.

Browse: See Select Text File

Save Information: To save the information provided above in the format of Matlab .mat files press this button and the accelerator key to perform this operation is **ALT+S**.

Cancel: To Cancel entering the other information about image and training set press cancel button but remember it is mandatory to save this information to perform **PCA** or **KL-Transform**.

** Image Width and Height should have the same value i.e. the square images should be used in Image Database or training set.

Select Text File Dialog Box

This is what the Image Database Browse Dialog looks like and it is popup in front of the Training Set Other Information Dialog Box

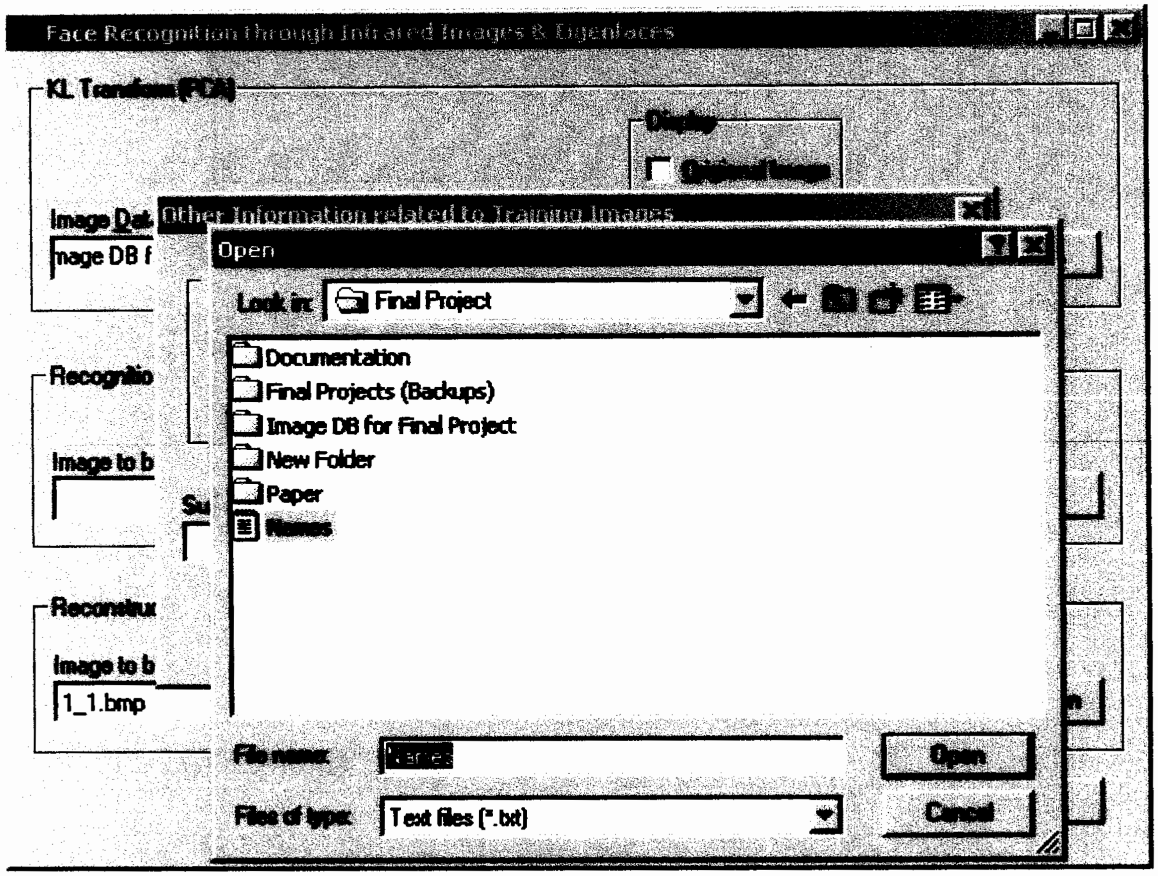


Figure 10. Select Subject Names File

Look in: It allow user to select the file of subject name. The accelerator key to focus this control is **ALT+I**.

File Name: it shows the file name which has been selected by the user as subject names file of training set.

Type of File: it shows the filter of files. we have only define the text filter so only the text file will be shown in the space provided under look in combo box.

Open: To open or select the file use this button. The accelerator key to perform this operation is **ALT+O**.

Cancel: To cancel the file selection press cancel button. and it will move u back to Training Set Other Information Dialog Box .

Eigenfaces:

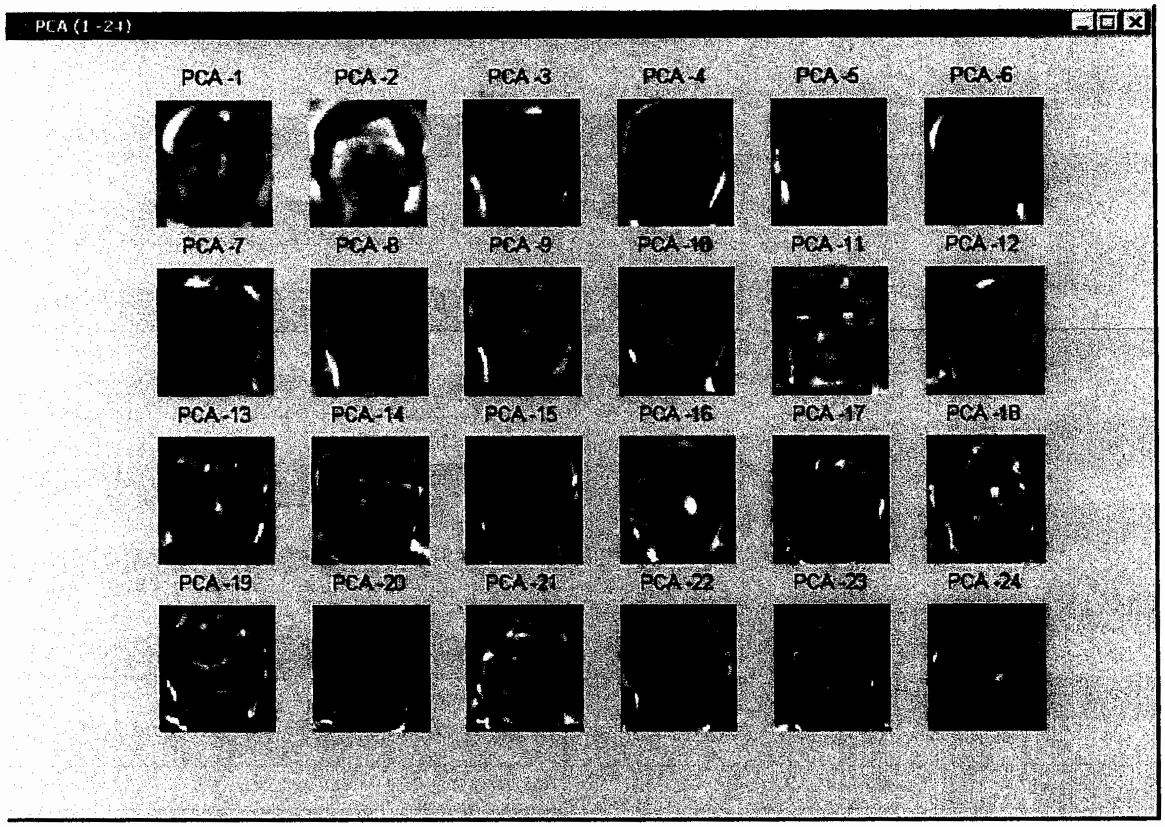


Figure 13. Eigenfaces

These are the eigenfaces formed depending on the eigenvalues (Graph of Eigenvalues) and 24 eigenfaces are shown in one figure and if the number of images is greater than 24 then next 24 eigenfaces are displayed in the next figure and so on.

Graph of Eigenvalues

This is what the Graph of Eigenvalues look like

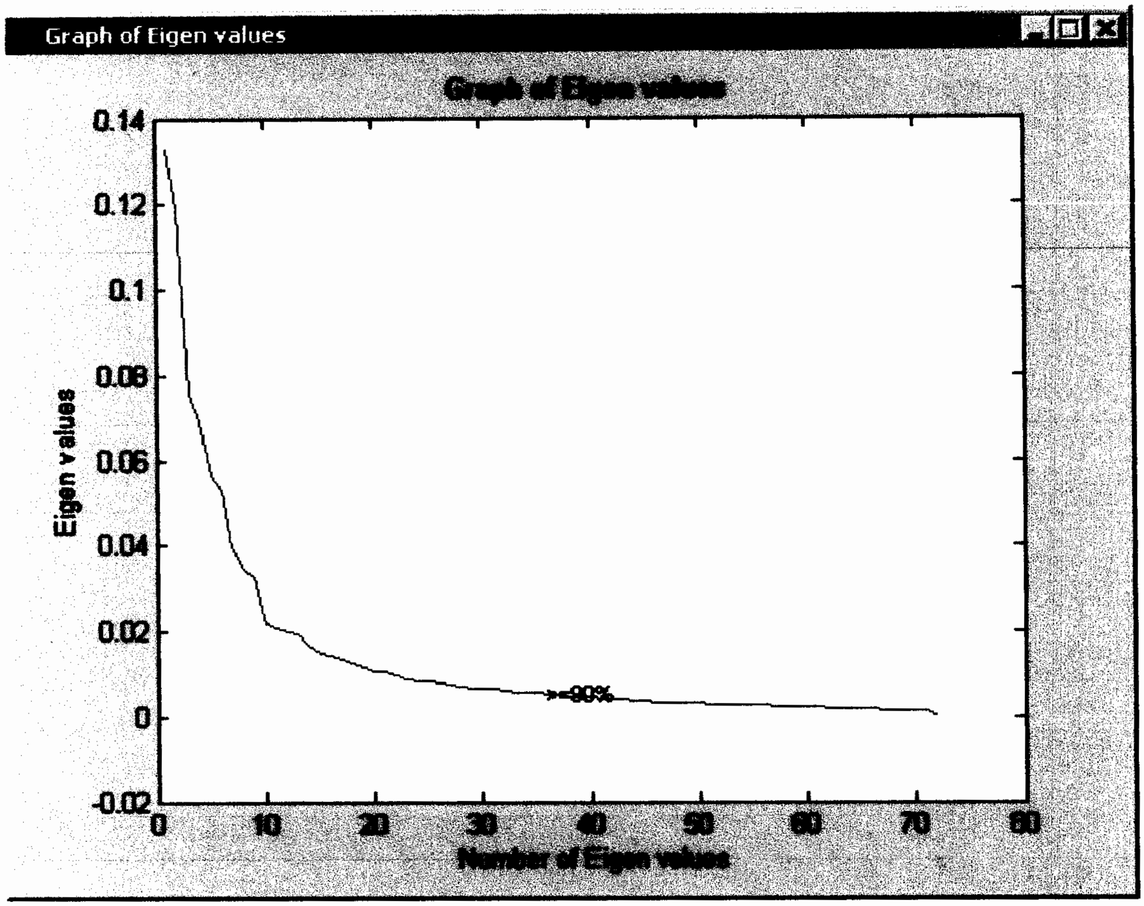


Figure 14. Graph of Eigenvalues Vs Number of Eigenvalues

It shows the Eigenvalues Vs Number of Eigenvalues and also place a Percentage or threshold which we have define on to tell the no of eigenvalues selected to make eigenspace the result Eigenfaces can be seen at <u>Sample Images</u>

Image to be Recognized Browse Dialog Box

This is what the Image Database Browse Dialog looks like and it is popup in front of the Main Window

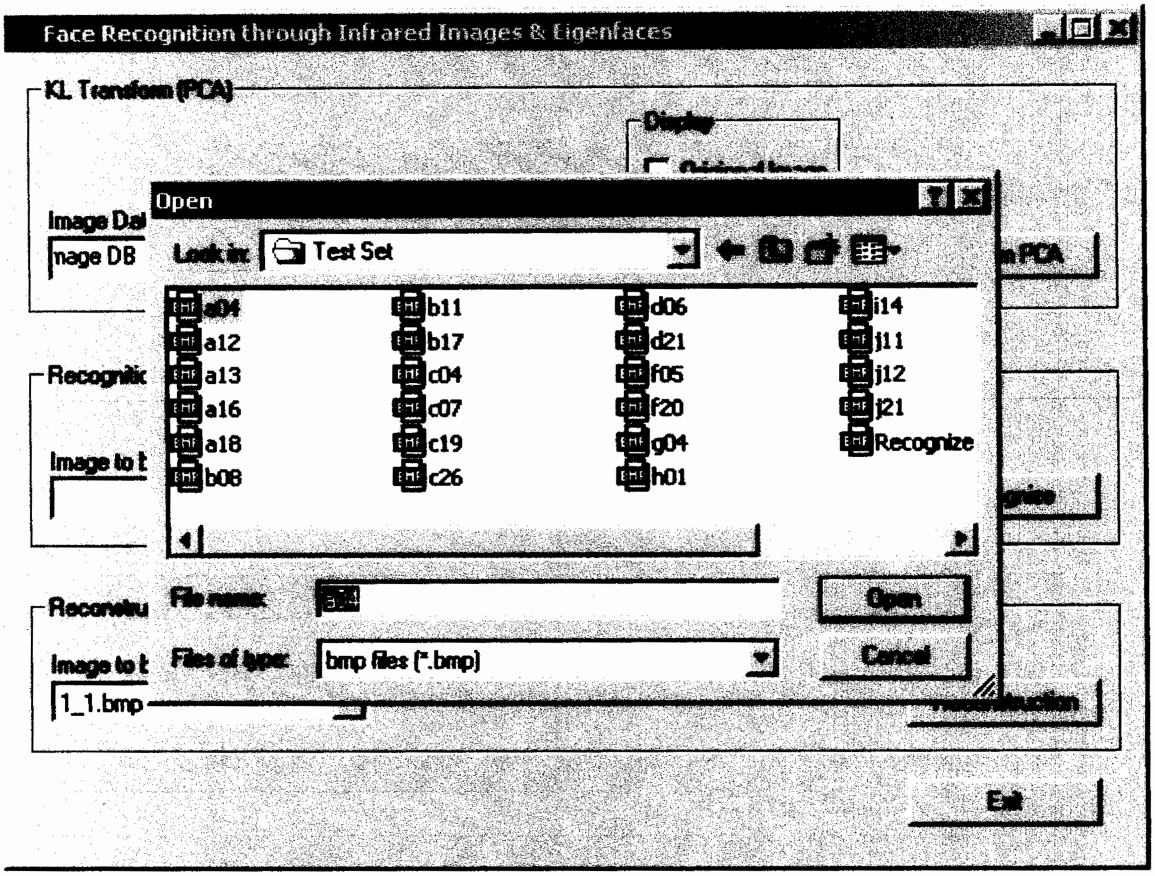


Figure 15. Select Image to be Recognized

Look in: It allow user to select the file of subject name. The accelerator key to focus this control is **ALT+I**.

File Name: it shows the file name which has been selected by the user as subject names file of training set.

Type of File: it shows the filter of files. we have only define the Bitmap filter so only .bmp file will be shown in the space provided under look in combo box.

Open: To open or select the file use this button. The accelerator key to perform this operation is ALT+O.

Cancel: To cancel the file selection press cancel button and it will move u back to Main Window .

Recognition Process Results

Here are some samples how the recognition process works:

Not a Face:

Figure 16. Not a face

This software first see whether or not image is a face, if not a face then the execution terminates and display the test image showing the title "Not a Face".

Not in Database:

Figure 17. Not In Database

When the match of image of a person not found then it display the image showing the title "Not in Database".

Recognition:

When test image is matched with database then the test image and the reconstructed Matched image are shown with test image containing title "Test Image" and Matched reconstructed image with the title "His Name" as shown in figure 18.

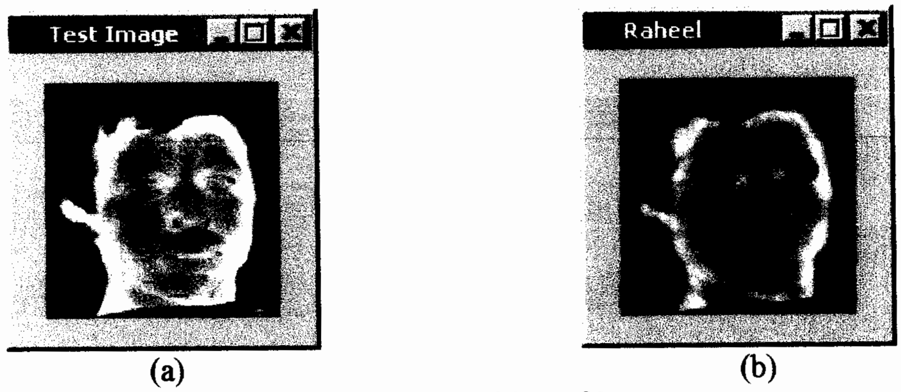


Figure 18. Recognition of Face

Recognition after Temperature effect Enhancement:

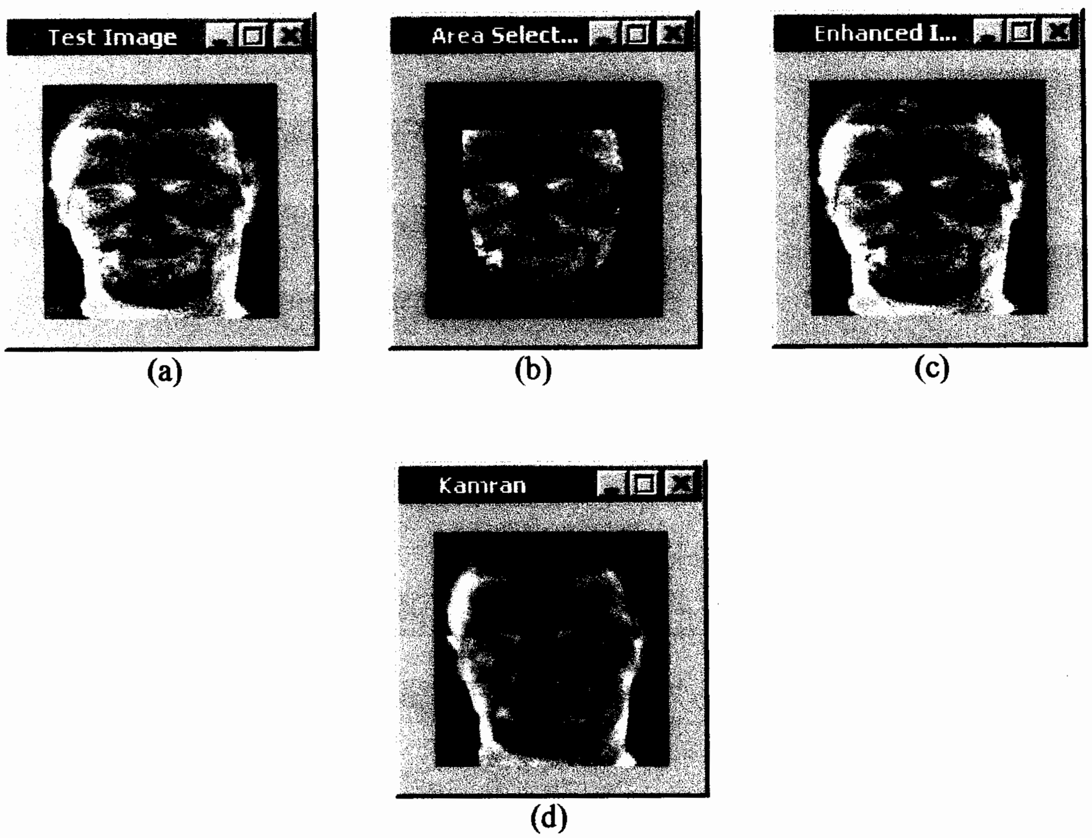


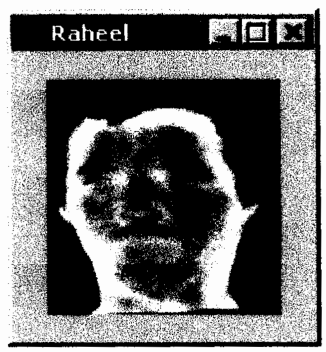
Figure 19. Recognition after passing through Enhancement Algorithm

When image need enhancement it is enhanced and if matched with database then Test Image , Selection Area, Enhanced Image and Matched reconstructed image are shown. and if not matched with database images then the first three images are shown with Test Image having title "Not in database"

Recognition after Sunglasses Filtering:

(a)

(b)



(c)

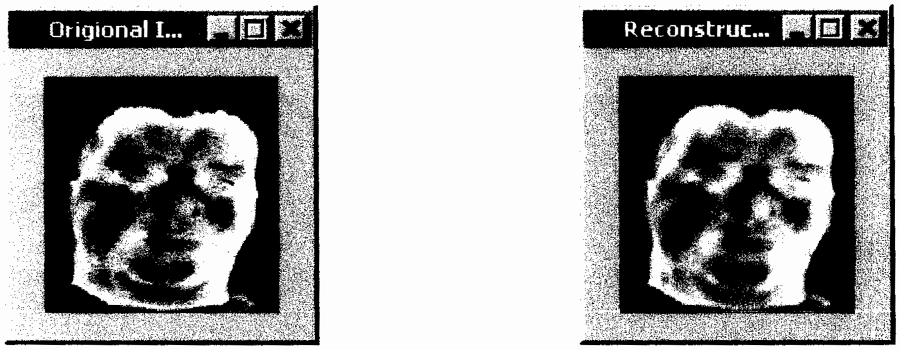
Figure 20. Recognition after passing through Sunglasses Filtering Algorithm

When image having sunglasses came its glasses are removed using algorithm and then matched with database and if found match then Test Image, enhanced Image, and Matched reconstructed Image are shown and if not matched then only Test and Enhanced Image are shown with Test Image having title "Not in Database".

Reconstruction Process Result

Here is a sample result of the Reconstruction Process.

Reconstruction:



(a) (b)

Figure 21. Reconstruction

Test Image along with the reconstructed image is shown. Here it can easily be seen that the reconstructed image is a little blur and different but it is due to we have selected almost half of the eigenvectors to define or create our eigenspace.

Software Development Team

Software Engineer:

Muhammad Raheel Siddiqui (m_raheel_siddiqui@yahoo.com)

Supervisors:

Prof. Dr. Khalid Rashid (drkhalid@iiu.edu.pk)

Muhammad Sher (msher313@yahoo.com)

Special Thanks to: See Acknowledgement

Acknowledgement

All Praise to the Almighty Allah, the most Merciful, the Most Gracious, without His help and blessings, I was unable to complete this project.

Thanks to my parents who helped me during my most difficult times and it is due to their unexplainable care and love that I am at this position today.

Thanks to my Project Supervisor Prof. Dr. Khalid Rashid and Muhammad Sher.

I Acknowledge Teachers and friends for their help in the project.

Muhammad Raheel Siddiqui

Contact Me

Muhammad Raheel Siddiqui

Email: tigerraheel@yahoo.com

m_raheel_siddiqui@yahoo.com

Mobile: 0300 5138396

Postal Address:

183/M-19 Quaidabad, Behind Zahid Market,

Dhoke Sayyedan, Misrial Road,

Rawalpindi.

Research Paper

Improved Face Recognition through Infrared Images and Eigenfaces after passing Test Face through Cold Effect Enhancement and Sunglasses Filtering Algorithms

Raheel Siddiqui   Muhammad Sher   Khalid Rashid

Department of Computer Science, International Islamic University, Islamabad, Pakistan.

tigerraheel@yahoo.com   msher313@hotmail.com   drkhalid@iiu.edu.pk

Abstract

This paper describes face identification using infrared images and eigenfaces after passing test face through cold effect enhancement and/or sunglasses filtering algorithms and handling facial hair through threshold. Eigenface technique after modification is used to define our eigenspace. Test image before going through the recognition process has to pass through a check to see whether it is a face image or not. The test face is passed through an algorithm to check and enhance if the person come from cold and then is projected to eigenspace to find the match. If match is not found then it is passed through another algorithm to check whether person has worn sunglasses and if so the image is enhanced in order to make recognition more efficient. Different expressions are used to increase the efficiency and check the impact of cold effect enhancement algorithm and sunglasses filtering algorithm. Only one person of the test sets is recognized wrongly with the other but both of them belongs to the training set and it gives 100% accurate results for profile images.

1. Introduction

Automated face recognition is a well-studied problem in computer vision [1]. Its current applications include security systems e.g. ATM's computer logins, secure building entrance, criminal photo and many others. Principal Component Analysis is one of the most successful techniques for face recognition and dimension reduction [7]. This technique is also known as eigenfaces.

Infrared images represent the heat pattern emitted from an object. Since the vein and tissue structure of a face is unique, the infrared images should also be unique [7]. In eigenface technique, we have training and test set of images. We compute eigenvectors of covariance matrix of the training set of images. Then we compute eigenfaces that are when displayed look like a ghostly faces and termed as eigenfaces. We can reconstruct any face in the training set by combining eigenfaces linearly. In this paper we compute the eigenfaces as described by Turk M. and Pentland [9] but with some modifications.

When test image come for recognition, it is checked whether this is a face in order to reduce the false acceptance rate because a non-face image after enhancement can gain some features of face then we pass it to the cold enhancement algorithm. The algorithm checks whether person has come from cold or not by checking pixel values at nose and cheeks. If the number of values is greater than the threshold then the selected area of face is enhanced and this selection is done by the algorithm it self and it enhances the values of pixel in that selected area. The test image is projected into the eigenspace and if it is below the specified upper threshold then the test image is of an authorized person. If the test image is not in the database then it passes through the sunglasses filtering algorithm and this algorithm checks whether a person whose image is taken wear glasses or not. If the number of values is greater than the threshold, the glasses area is enhanced and the pixel values are replaced with the nose tip pixel value. Again it is projected to eigenspace and if the error is below the specified threshold then the face is of an authorized person.

2. Training Set to Define Eigenspace

The Training set can contain any number of images greater than one because eigenvector can not be zero. To support recognition of different expression of same face, the training set contains equal number of expressions for all subjects. Our training set contains the same number of images for every subject but the expressions may vary for different subjects. For example, there is a smiley image for one subject and other subject contains an angry expression image. Training set cannot contain any image for subject wearing sunglasses because experiment shows that it leads to a very high False Acceptance Rate, which can not be afforded in security systems. Experiments also show that training set should not have image of person wearing sunglasses because it leads toward high False Acceptance and False Rejection.

3. Principle Component Analysis

Consider face images of size n x n. Those images can be thought of as a vector of dimension n^2 or point in n^2 dimensional space and therefore the training set of images corresponds to a set of points in the high dimensional space. Since the facial images are similar in structure and can be described by lower dimensional subspace. We get the basic vectors each of length n^2 through Principal Component Analysis.

Let $I_1, I_2, \ldots, I_N$ be images of the training set then mean image is calculated by

$$MI = (\Sigma_{j=1\ldots N} Ij)/N$$

where N is the total no of images.

The mean deviation is calculated through

$$Xj = Ij - MI$$

Mean deviation is also called normalized face-vector and written as.

$$\Phi = [X_1\ X_2 \ldots X_N]$$

The covariance matrix is

$$C = (\Phi.\Phi')/N$$

Having dimension $n^2 \times n^2$.

Calculating eigenvectors of C for size n^2 by n^2 is intractable so we determine the eigenvectors by solving an N x N matrix instead [2][9]. Highest corresponding M eigenvectors are chosen to span an M-dimensional subspace.

We have introduced two more steps and altered one step to make the threshold selection and calculation easy. We have normalized the mean deviation vector and eigenspace by their norms and have not multiplied the covariance matrix by factor 1/N. The working of the algorithm is shown in Figure 1.

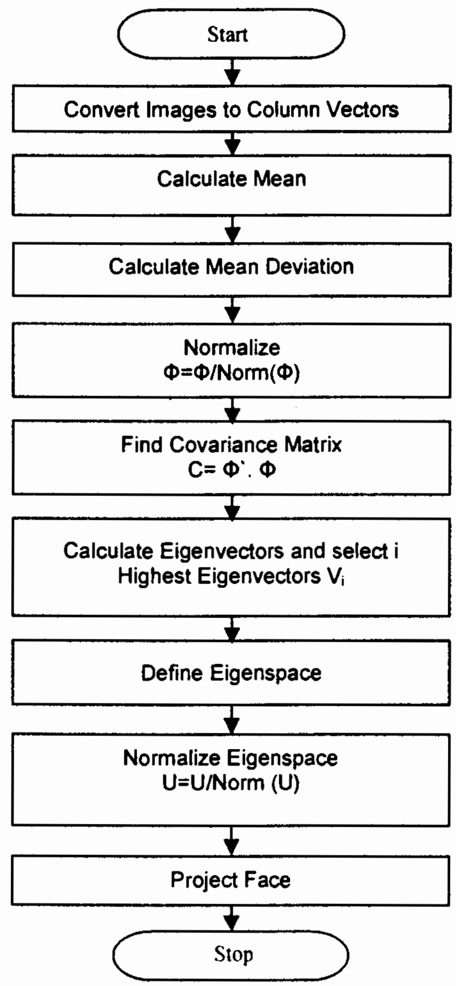


Figure 1. Working of PCA to define our Eigenspace

We project a test image onto eigenspace using following Operations

$$w_i = u_i' * (TestImage - MI)$$

Where i=1, 2, …., M

Where w_i are the weights and forms a vector $[w_1\ w_2\ w_3\ ...\ w_M]$ where each weight shows the contribution of each Eigenface in representing the test face image [7][9][6] and u_i are the i highest eigenvectors corresponding to i highest eigenvalues. We check whether it is a face or not in order to increase the efficiency and reduce the False Acceptance Rate because after passing through algorithm it can be possible that non-face image gain some features of face.

4. Cold Effect Enhancement Algorithm

The face of a person has an impact of cold when he comes from cold especially the nose and cheeks area is highly sensitive to cold. Infrared images are the heat patterns emitted by an object [7]. When person coming from cold stands in front of infrared camera, infrared camera catches this impact. This image will be different from the normal image. Hence it needs to be enhanced before recognition process.

Previously *Rose Cutler* has done limited work on face recognition using infrared images and has not covered this problem [7].

To overcome this problem we have developed an algorithm to enhance the face area of images. If the test image is a face then it passes through enhancement algorithm where we check the nose and cheeks to see whether person comes from cold or not. This area has been chosen to check because the nose and cheeks are mainly affected by cold. If it attains a value greater than a specified threshold then selected area of the face is enhanced and the values in that area are normalized. Figure 2 shows graphically how this algorithm works.

Figure 2. Result after passing through cold enhancement algorithm

The image in Figure 2(a) is the actual image came for enhancement and Figure 2(b) shows the selection area and Figure 2(c) shows the image after enhancement. It is enhanced such that it must not lead to False Acceptance or False Rejection. This algorithm is not dimension dependent. It means that the face can be of any dimension but with a restriction that test image should be a square image. After enhancement it is again projected into face space and if the error is under the threshold, the person is considered an authorized person. The working of cold effect enhancement algorithm is shown in Figure 3.

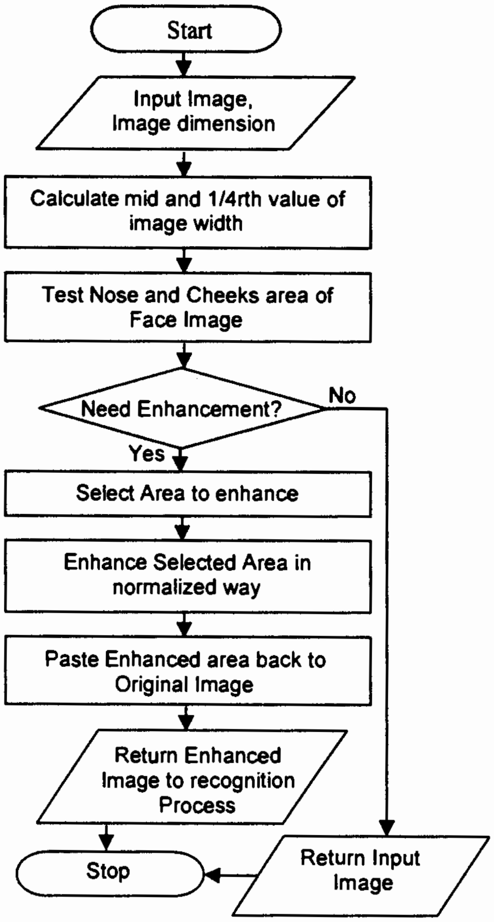


Figure 3. Working of Cold Effect Enhancement Algorithm

5. Sunglasses Filtering Algorithm

There is a possibility that a person has worn sunglasses when he came in front of camera. The image in this case will have very small pixel values and the area of sunglasses in the image will appear black. The solution to this problem is necessary because it have more importance when the camera is placed in the public places to pick the required person by the police. Limited work is done in this area for infrared images [7].

To solve this problem we have suggested a solution in which the normal glasses problem is solved through threshold and an algorithm is developed to filter the sunglasses in normalized way. If error after cold effect enhancement is above the specified threshold the enhanced test

.mage is passed to the sunglasses enhancement algorithm. The algorithm checks that whether there are sunglasses or not and if found then it replaces the sunglasses with the nose tip pixel value in order to normalize it. The nose tip pixel is chosen because it is the part of the face first affected by the cold so that it should not assign a value to the area of sunglasses that will increase the False Acceptance Rate. We cannot make eyes of that person because it may lead to False Acceptance. This algorithm is also dimension independent and restriction is that image must be a square image.

(a) (b)

Figure 4. Result after passing through sunglasses filtering algorithm

Figure 4(a) is the original image came for recognition and Figure 4(b) is the image after enhancement. We can't exactly guess which kind of eyes this person has that's why we have not created eyes in order to avoid the False Acceptance. Figure 5 describes working of sunglasses filtering algorithm. It shows the whole procedure how the test image is processed to check and filter the area of sunglasses if needed.

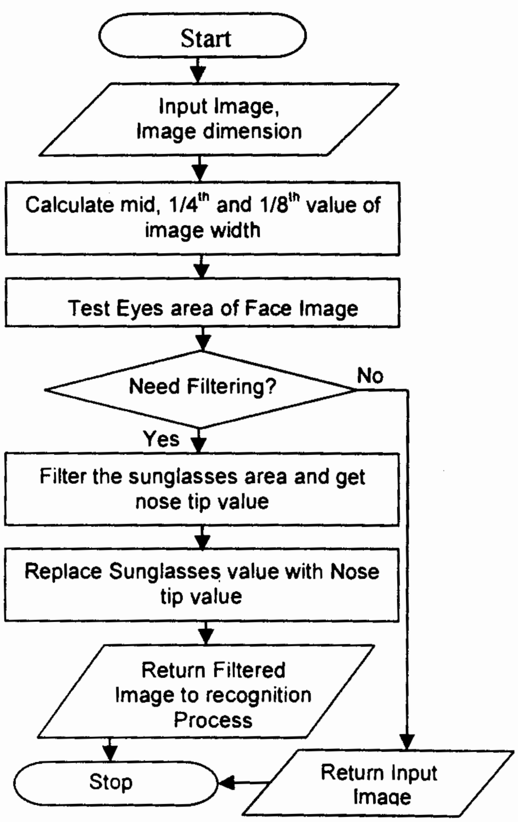


Figure 5. Working of Sunglasses Filtering Algorithm

6. Recognition Process

Figure 6 shows the whole recognition process from where test image come and then normalize. After normalizing it passes through enhancement algorithm. If needs, enhancement is done and then matched with database and if it matches then the image is reconstructed using eigenfaces.

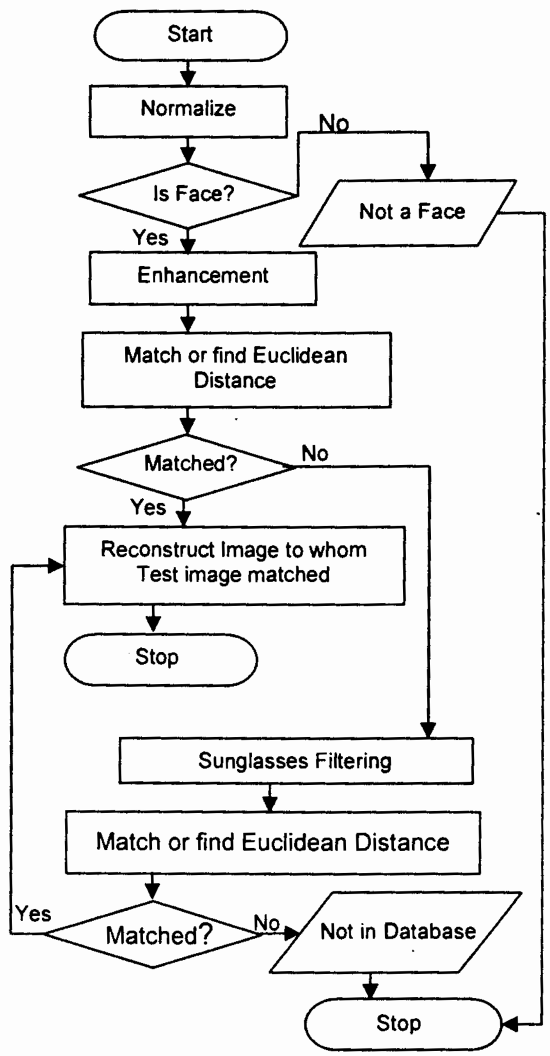


Figure 6. Recognition Process

If it is not matched then it passes through sunglasses Algorithm. In case a face image wearing glasses is found, it is enhanced and glasses pixel values are replaced with nose tip pixel value and then again match with database and if matched then the image is reconstructed using eigenfaces and if not found then message is shown "Not in database". The reason to replace the glasses value with value from nose is to make the enhancement in normalized way so that it should not lead us to false acceptance rate

7. Results

We have used two training sets and two test sets to evaluate the working of our algorithms. Our first training set consists of 4 subjects having 1 expression for each person. Sample infrared images are shown in Figure 7. The dimension of images is 117 X 117 but is not fixed; training set can have any square dimensions. The alignment of the faces is made manually. Mean image of the training set is calculated and is shown in Figure 8.

(a) (b) (c) (d)

Figure 7. Infrared images of training set of 4 images

Figure 8. Mean image

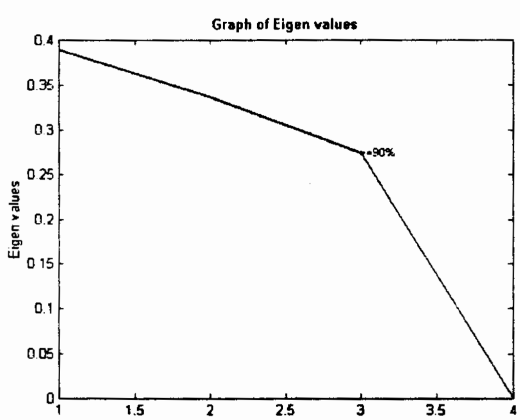


Figure 9. Graph of Eigenvalues vs Number of Eigenvalues

Graph of eigenvalues vs numbers of eigenvalues show the selection eigenvectors corresponding to highest eigenvalues to define our eigenspace.

(a) (b) (c)

Figure 10. Principal Components

Sample Images of test set that are projected onto eigenspace in order to find a match are shown in Figure 11.

(a) (b) (c) (d)

Figure11. Sample Infrared images from test set

When projecting an image onto eigenspace the distance is checked from eigenspace and if the distance is greater than the defined face image threshold, image is consider as non-face image. If the distance is greater than the second threshold define to find match then the projected image is consider as face but not of a person from the training set else the image is reconstructed from eigenspace to which the projected image is matched. Sample results are shown in Figure 12, Figure 13 and Figure 14 respectively. Figure 14(a) is the projected image, Figure 14(b) shows the filtered Image and Figure 14(c) is the reconstructed image to whom projected image matched.

Figure 12. Not a face

Figure 13. Not in database

(a) (b) (c)

Figure 14. Match is found

If image is not a face or not in database, projected image is displayed with title "not a face" or "not in database". The matched image is reconstructed if match is found after projection of test image.

Our second training sets consist of 8 subjects having 9 expressions each that makes total of 72 images. Sample infrared images are shown in Figure 15. The expressions are not the same for each subject but number of expressions is the same. The dimension of images is 117 X 117 but it is not fixed training set can have any square dimensions. The alignments of the faces are made manually. Mean image of the training set images is calculated and is shown in Figure 16.

(a) (b) (c) (d)

Figure15. Sample infrared images from the training set of 72 images

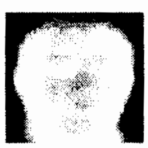


Figure 16. Mean image of 72 images

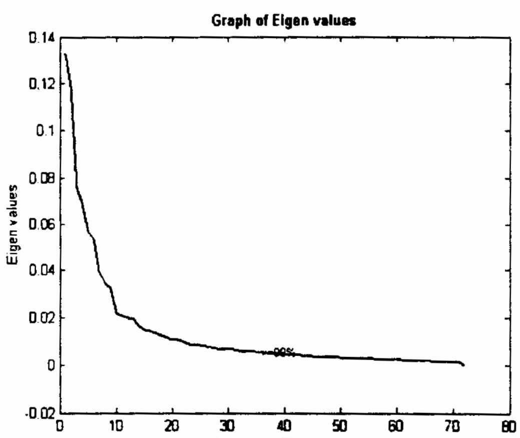


Figure 17. Graph of Eigenvalues vs Number of Eigenvalues

Graph of eigenvalues vs numbers of eigenvalues show the selection of 36 eigenvectors corresponding to 36 highest eigenvalues to define our eigenspace.

(a) (b) (c)

Figure 18. First three Principal components

Our test set contains the images of the subjects having different expressions used in our training set and image of persons that are not in our training set.

(a) (b) (c) (d)

Figure 19. Sample images from the test set

Figure 20, Figure 21, Figure 22 and Figure 23 shows the sample results obtained after projecting test images from the test set to eigenspace. Figure 22(a) is the projected or test image and Figure 22(b) is the reconstructed image to whom projected image matched. The reconstructed image is blur because of the selection of almost half highest corresponding eigenvectors.

Figure 20. Not a face

Figure 21. Not in database

(a) (b)

Figure 22. Match is found

Only one face is recognized wrongly with other but both the person belongs to the training set. Figure 23(a) wrongly matches with Figure 23(b) but both persons belong to our training set.

(a)

(b)

Figure 23. Result of the test image

Rose Cutler has used 2 expressions (normal and smiley) in the training set where as we have used 9 expressions to implement our algorithms. Accuracy obtained by Rose Cutler was 96.6% where as our approach has shown accuracy of 99%.

8. Conclusion

The results show that our algorithms for cold effect enhancement and sunglasses filtering works quite well and do not lead to a false acceptance. These algorithms are developed to use in the security systems but they can be used in recognizing the person in public places and can help to pick the required person by the police. The Future work will involve the automated alignment of the face. The higher resolution images can produce better results. Our algorithm of PCA can be extended to motion detection, object recognition.

9. References

[1] Chellapa. R., et. Al. "Human and Machine Recognition of Faces: A Survey". UMCP CS-TR-3339. 1994.

[2] K. Fukunage. "Introduction to Statistical Patern Recognition" (2nd edition), Academic Press. New York. 1990.

[3] Moghaddam and Pentland. "Face Recognition using View-Based and Modular Eigenspaces. Automatic Systems for the Identifications and Inspection of Humans". SPIE Vol. 2277

[4] M. Kirby and L. Sirovich. "Application of the Karhunen-Loeve procedure for the characterization of human faces". IEEE Transaction on pattern Analysis and Machine Intelligence. Vol 12. no. 1. pp. 103-108. January 1990.

[5] M. Turk. A. Pentland. "Face Recognition using Eigenfaces". IEEE Conf. on Computer Vision and Pattern Recognition. 1991.

[6] Pentland, et. Al. "Experiment with Eigenfaces". MIT VISMOD TR-194

[7] Rose Cutler, "Face Recognition Using Infrared Images and Eigenfaces", 1996.

[8] Steve, Peter and Ingemar, "Face Recognition using Mixture-Distance and Raw Images", IEEE International Conference on Systems, 1997.

[9] Turk, M., and Pentland A., "Eigenfaces for Recognition", Journal of Cognitive Neuroscience. Vol.3, No 1, pp. 71-86, 1991.

[10] T. Fromhez, "Face Recognition: A summary of 1995-1997". Technical Report ICSI TR-98-027. International Computer Science Institute, Berkeley. 1998.

[11] U. Grenander. A. Srivastava and C Hesher. 01] "Models for Statistical Analysis of Range" Images, November 2001.