ABSTRACT

We propose an facial affirmation system called Lapland affirmation. Using the location Defense projections , face images assigned to subspace of faces to examine. It is not exactly equivalent to the main component The evaluation PCA & the discriminant linear analysis LDA that observe only the structure of facial space. This finds entering the information from the blocked neighborhood and gaining a face subspace that better recognizes the complex structure of the main face. The Laplacian faces are the perfect approximations directed to functions of manager of Laplace in the facial complex. In therefore, the annoying assortments that arise due to changes in lighting, external appearance and posture can be discarded or reduced.

Speculative examination shows that PCA as well as LDA along with LPP can be derived from various models of the graphic. Let's consider proposal Focus laplacian face along with the Eigen face and also Fisher face systems in three assorted educational assortments. The test results suggest that Laplacian face proposed approach offers unmatched representation and gets less failure rates of face affirmation.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

A smart environment is one that is able to identify people, interpret their actions, and react appropriately. Thus, one of the most important building blocks of smart environments is a person identification system. Face recognition devices are ideal for such systems, since they have recently become fast, cheap, unobtrusive, and, when combined with voice-recognition, are very robust against changes in the environment. Moreover, since humans primarily recognize each other by their faces and voices, they feel comfortable interacting with an environment that does the same.

Facial recognition systems are built on computer programs that analyze images of human faces for the purpose of identifying them. The programs take a facial image, measure characteristics such as the distance between the eyes, the length of the nose, and the angle of the jaw, and create a unique file called a "template." Using templates, the software then compares that image with another image and produces a score that measures how similar the images are to each other. Typical sources of images for use in facial recognition include video camera signals and pre-existing photos such as those in driver's license databases.

Facial recognition systems are computer based security systems that are able to automatically detect and identify human faces. These systems depend on a recognition algorithm, such as eigenface or the hidden Markov model. The first step for a facial recognition system is to recognize a human face and extract it fro the rest of the scene. Next, the system measures nodal points on the face, such as the distance between the eyes, the shape of the cheekbones and other distinguishable features. These nodal points are then compared to the nodal points computed from a database of pictures in order to find a match. Obviously, such a system is limited based on the angle of the face captured and the lighting conditions present. New technologies are currently in development to create three-dimensional models of a person's face based on a digital photograph in order to create more nodal points for comparison. However, such technology is inherently susceptible to error given that the computer is extrapolating a three-dimensional model from a two-dimensional photograph.

Principle Component Analysis is an eigenvector method designed to model linear variation in high-dimensional data. PCA performs dimensionality reduction by projecting the original n-dimensional data onto the $k << n$ -dimensional linear subspace spanned by the leading eigen vectors of the data's covariance matrix. Its goal is to find a set of mutually orthogonal basis functions that capture the directions of maximum variance in the data and for which the coefficients are pairwise decorrelated. For linearly embedded manifolds, PCA is guaranteed to discover the dimensionality of the manifold and produces a compact representation.

Facial Recognition Applications

Facial recognition is deployed in large-scale citizen identification applications, surveillance applications, law enforcement applications such as booking stations, and kiosks

1.1 Problem Definition

Facial recognition systems are computer-based security systems that are able to automatically detect and identify human faces. These systems depend on a recognition algorithm. But the most of the algorithm considers some what global data patterns while recognition process. This will not yield accurate recognition system.

So we propose a face recognition system which can able to recognition with maximum accuracy as possible.

1.2 System Environment

The front end is designed and executed with the J2SDK1.4.0 handling the core java part with User interface Swing component. Java is robust , object oriented , multi-threaded , distributed , secure and platform independent language. It has wide variety of package to implement our requirement and number of classes and methods can be utilized for programming purpose. These features make the programmer's to implement to require concept and algorithm very easier way in Java.

The features of Java are as follows:

Core java contains the concepts like Exception handling, Multithreading, Streams can be well utilized in the project environment. The Exception handling can be done with predefined exception and has provision for writing custom exception for our application. Garbage collection is done automatically, so that it is very secure in memory management. The user interface can be done with the Abstract Window tool Kit and also Swing class. This has variety of classes for components and containers. We can make instance of these classes and this instances denotes particular object that can be utilized in our program.

Event handling can be performed with Delegate Event model. The objects are assigned to the Listener that observe for event, when the event takes place the corresponding methods to handle that event will be called.

This application make use of ActionListener interface and the event click event gets handled by this. The separate method actionPerformed() method contains details about the response of event.

System Requirements

Hardware specifications

Processor	:	Intel Processor IV
RAM	:	128 MB
Hard disk	:	20 GB
CD drive	:	40 x Samsung
Floppy drive	:	1.44 MB
Monitor	:	15’ Samtron color
Keyboard	:	108 mercury keyboard

Software Specification

Operating System	:	Windows XP/2000
Language used	:	J2sdk1.

CHAPTER 2
SYSTEM ARCHITECTURE

2.1 Existing System

Facial recognition systems are computer-based security systems that are able to automatically detect and identify human faces. These systems depend on a recognition algorithm. Principal Component Analysis (PCA) is a statistical method under the broad title of factor analysis. The purpose of PCA is to reduce the large dimensionality of the data space (observed variables) to the smaller intrinsic dimensionality of feature space (independent variables), which are needed to describe the data economically. This is the case when there is a strong correlation between observed variables. The jobs which PCA can do are prediction, redundancy removal, feature extraction, data compression, etc. Because PCA is a known powerful technique which can do something in the linear domain, applications having linear models are suitable, such as signal processing, image processing, system and control theory, communications, etc.

The main idea of using PCA for face recognition is to express the large 1-D vector of pixels constructed from 2-D face image into the compact principal components of the feature space. This is called eigenspace projection. Eigenspace is calculated by identifying the eigenvectors of the covariance matrix derived from a set of fingerprint images(vectors).

But the most of the algorithm considers some what global data patterns while recognition process. This will not yield accurate recognition system because

- Less accurate
- Does not deal with manifold structure
- It doest not deal with biometric characteristics.

2.2 Proposed System

PCA and LDA aim to preserve the global structure. However, in many real-world applications, the local structure is more important. In this section, we describe a Locality Preserving Projection (LPP) [9], a new algorithm for learning a locality preserving subspace. The complete derivation and theoretical justifications of LPP can be traced back to [9]. LPP seeks to preserve the intrinsic geometry of the data and local structure. The objective function of LPP is as follows.

LPP is a general method for manifold learning. It is obtained by finding the optimal linear approximations to the eigenfunctions of the Laplace Betrami operator on the manifold [9]. Therefore, though it is still a linear technique, it seems to recover important aspects of the intrinsic nonlinear manifold structure by preserving local structure. Based on LPP, we describe our Laplacianfaces method for face representation in a locality preserving subspace.

In the face analysis and recognition problem, one is confronted with the difficulty that the matrix XDXT is sometimes singular.

This stems from the fact that sometimes the number of images in the training set ðnÞ is much smaller than the number of pixels in each image ðmÞ. In such a case, the rank of XDXT is at most n, while XDXT is an m _ m matrix,

which implies that XDXT is singular. To overcome the complication of a singular XDXT , we first project the image set to a PCA subspace so that the resulting matrix XDXT is nonsingular. Another consideration of using PCA as preprocessingis for noise reduction.

This method, we call Laplacianfaces, can learn an optimal subspace for face representation and recognition. The algorithmic procedure

of Laplacianfaces is formally stated below:

1. **PCA projection** : We project the image set fx into the PCA subspace by throwing away the smallest principal components. In our experiments, we kept 98 percent information in the sense of reconstruction error. For the sake of simplicity, we still use x to denote the images in the PC A subspace in the following steps .We denote by WPCA the transformation matrix of PCA.

2. Constructing the nearest-neighbor graph. Let G denote a graph with n nodes. The ith node corresponds to the face image xi. We put an edge between nodes i and j if xi and xj are “close,” i.e., xj is among k nearest neighbors of xi, or xi is among k nearest neighbors of xj. The constructed nearest neighbor graph is an approximation of the local manifold structure. Note that here we do not use the "-neighborhood to construct the graph. This is simply because it is often difficult to choose the optimal " in the real-world applications, while k nearest-neighbour graph can be constructed more stably. The disadvantage is that the k

nearest-neighbor search will increase the computational complexity of our algorithm.When the computational complexity is a major concern, one can switch to the "-neighborhood.

3. Choosing the weights. If node i and j are connected, put

column (or row, since S is symmetric) sums of S is the Laplacian matrix. The ith row of matrix X is xi. be the solutions of (35), ordered according to their eigenvalues.

These eigenvalues are equal to or greater than zero because the matrices XLXT and XDXT are both symmetric and positive semidefinite. Thus, the embedding is as follows:

where y is a k-dimensional vector. W is the transformation matrix. This linear mapping best preserves the manifold's estimated intrinsic geometry in a linear sense. The column vectors of W are the so-called Laplacianfaces.

This principle is implemented with unsupervised learning concept with training and test data.

2.3 Functional Requirement:

The system must require to implement Principle Component Analysis to reduce image in the dimension less than n and co-variance of the data. The system must be used in Unsupervised learning algorithm . So it must be trained properly with relevant data sets.

Based on this training input data is tested by the application and result is displayed to the user.

2.4 System Analysis Methods

System analysis can be defined, as a method that is determined to use the resources, machine in the best manner and perform tasks to meet the information needs of an organization. It is also a management technique that helps us in designing a new systems or improving an existing system. The four basic elements in the system analysis are

- Output
- Input
- Files
- Process

The above-mentioned are mentioned are the four basis of the System Analysis.

Feasibility Study

Feasibility is the study of whether or not the project is worth doing. The process that follows this determination is called a Feasibility Study. This study is taken in right time constraints and normally culminates in a written and oral feasibility report. This feasibility study is categorized into seven different types. They are

- Technical Analysis
- Economical Analysis
- Performance Analysis
- Control and Security Analysis
- Efficiency Analysis
- Service Analysis

Technical Analysis

This analysis is concerned with specifying the software that will successfully satisfy the user requirements. The technical needs of a system are to have the facility to produce the outputs in a given time and the response time under certain conditions..

Economic Analysis

Economic Analysis is the most frequently used technique for evaluating the effectiveness of prepared system. This is called Cost/Benefit analysis. It is used to determine the benefits and savings that are expected from a proposed system and compare them with costs. If the benefits overweigh the cost, then the decision is taken to the design phase and implements the system.

Performance Analysis

The analysis on the performance of a system is also a very important analysis. This analysis analyses about the performance of the system both before and after the proposed system. If the analysis proves to be satisfying from the company's side then this analysis result is moved to the next analysis phase. Performance analysis is nothing but invoking at program execution to pinpoint where bottle necks or other performance problems such as memory leaks might occur. If the problem is spotted out then it can be rectified.

Efficiency Analysis

This analysis mainly deals with the efficiency of the system based on this project. The resources required by the program to perform a particular function are analyzed in this phase. It is also checks how efficient the project is on the system, in spite of any changes in the system. The efficiency of the system should be analyzed in such a way that the user