enVisage: Face Recognition in Videos

Ashwin Venkatraman
(av102@doc.ic.ac.uk)

Supervisor: Dr. Stefan Rüeger
Second Marker: Ian Harries

June 14, 2006
Abstract

Face Recognition Technology for many years had been focussed on one face: frontal, well illuminated and looking directly at the capture device. In reality we have many faces in the eyes of others. As we move, interact and carry out our daily tasks, our face is being viewed in a variety of poses, under a range of illumination conditions and changing expression.

In recent years, there has been much attention placed on dealing with each of these forms of variation individually. However combining techniques to deal with multiple forms of variation at once, has received less scrutiny.

This project is primarily an investigation into forming an algorithm for identifying an individual with changes in pose and illumination. I utilise a number of established techniques which cater for these forms of variation individually, and unify them into a single novel algorithm.

enVisage, the application implemented as part of this project, addresses and unifies all aspects of a recognition system, and can be used to automatically recognise faces in both still and video sequences.
Acknowledgements

I would like to express my thanks and gratitude to:

• My supervisor, Dr. Stefan Rüeger, for his invaluable guidance and support throughout this project

• My second marker, Ian Harries for his constructive comments on the project and his thoughts on computing.

• My parents for their love and support during this challenging time.
Contents

1 Introduction 7
  1.1 Motivation .................................................. 7
  1.2 Objectives ................................................... 8
  1.3 Achievements ............................................... 9
  1.4 Report Structure .......................................... 9

2 Background 11
  2.1 Face Recognition - an Overview .......................... 11
  2.2 Face Detection & Tracking ................................. 13
  2.3 Feature Extraction ......................................... 14
  2.4 Invariant Recognition to Pose Variation .................. 21
  2.5 Invariant Recognition to Illumination Variation ........... 24
  2.6 Selection of Frames for Recognition ....................... 26

3 Face Detection & Tracking 29
  3.1 Detection .................................................... 29
  3.2 Tracking ..................................................... 33
  3.3 Unifying Detection & Tracking ............................. 34
  3.4 Manual Annotation .......................................... 36

4 Design of a Novel Algorithm for Identification 37
  4.1 The Problem .................................................. 37
  4.2 The Solution ................................................ 38
  4.3 Summary ..................................................... 41

5 Implementation of a Novel Algorithm for Identification 43
  5.1 Eigenfaces .................................................... 43
  5.2 Bayesian Eigenfaces ......................................... 48
  5.3 View Based Eigenfaces ...................................... 50
  5.4 Face Normalisation ......................................... 51

6 Application To Video 53
  6.1 Overview ..................................................... 53
  6.2 A Model for Identification ................................ 53
Chapter 1

Introduction

As humans we have the inept ability to perform face recognition routinely and effortlessly as part of our daily lives. However automating this task is an extremely challenging problem, and with over 30 years of research it is still largely regarded as unsolved.

1.1 Motivation

In recent years, there has been a resurgence of interest in face recognition as a result of the availability of feasible technologies and a growing requirement for automatic human identification in numerous applications.

In a post 9-11 era, increased demands in security have resulted in immense growth in research initiatives of biometric technologies. The purpose of such applications are to utilise some biological characteristic to be able to automatically identify an individual.

As a result, face recognition has been drawn to the forefront, becoming one of the primary biometric technologies. It has a clear advantage over other technologies as it is natural, non-intrusive, and discrete (i.e. the subject need not be aware or take any specific action to be identified).

However the above advantage is to a certain extent a double-edged sword. As the subject is unconstrained and can act naturally, the system has to cater for the inherent variation (e.g. pose, expression etc.) in the manner in which they may present themselves.

For example, it is of little use if a system can only identify frontal images of the face, as the subject may only appear in a non-frontal pose. Thus whilst the nonintrusive nature of face recognition is advantageous, it also creates an immense challenge, one which is yet to be solved.
If face recognition technology as a means of security is to come into fruition, this is a challenge it must ultimately overcome. It goes without saying, that further research and investigation in the area is much required, which forms as the main motivation for our project.

1.2 Objectives

The term *Face Recognition* is a deceiving one. It gives the impression that a single algorithm that identifies faces, would be a solution to this problem. This is not the complete story.

Although identification is indeed an essential part of the process required to recognize an individual, there are other interdependent components, (which have been and could be projects in their own right) which need to be addressed if automated recognition is to be achieved. This distinction is made clear in Chapter 2.

The overall objective of this project was to produce a system that can recognize known faces in still images and video probes.

However it was not feasible to address each of these components in detail, to produce such a system in the time given. Therefore the primary focus has been to implement a novel recognition algorithm that is invariant to pose and illumination, by integrating methods specified in current research.

The remaining components required by our system have been accomplished using available libraries, or by providing manual means via the user interface.

To summarise, the primary objectives of this project were to:

- Implement a novel algorithm for identifying an individual, subject to changes in pose and illumination
- Implement an algorithm to segment faces from video/still images, using open source libraries
- Engineer software to unify the above two components, into a recognition system

*Ideally, using the software developed, a user would be able to load an image/video file, and with a single click the system should be able to recognise all individuals in the scene it has knowledge of.*
1.3 Achievements

On completion, the main contributions made by the project are:

- A multi-view face detector, which is able to detect faces with out of plane rotation ranging from $-90^\circ$ to $+90^\circ$, with detection rates of 49% for profile views, and 73% for non frontal images.

- An novel identification algorithm reporting, 60% and 73% identification rate, for variations in pose and illumination respectively.

- Implementation of enVisage the software, integrating segmentation and identification to provide automated recognition in still images and video sequences.

1.4 Report Structure

The report is set out as described below:

- *Chapter 2: Background*, gives an overview of the recognition problem and the various components it comprises of. Detailed literature review of techniques relating to the identification algorithm are provided, whilst other areas of recognition are also touched upon.

- *Chapter 3: Face Detection & Tracking*, provides a detailed analysis of the technology behind the libraries used for segmenting faces. In addition implementation details as to how the libraries were unified and extended are also provided.

- *Chapter 4: Design of a Novel Identification Algorithm*, outlines the specific aims of identification algorithm. It then goes onto provide a blueprint for the novel algorithm to be implemented.

- *Chapter 5: Implementation of a Novel Identification Algorithm*, contains detailed information relating to the various components making up the Identification algorithm, and any considerations which were made when implementing them.

- *Chapter 6: Application to Video*, describes how the system was extended from still images to video.

- *Chapter 7: Engineering enVisage*, consists of the design of the software which unifies the technology described in the previous chapters, to form our recognition system. Design of the architecture and user interface, as well as description of the technology and methods used are provided.
• **Chapter 8: Testing & Evaluation**, provides details of the testing carried out to measure the performance of the identification & face detection algorithms.

• **Chapter 9: Conclusion**, summarises the contributions of this project and outlines possible future research work.
Chapter 2

Background

The following chapter provides an overview of the problem we are faced with, and a review of the state of the art techniques that have been applied to solve it.

2.1 Face Recognition - an Overview

The overall goal of a face recognition system can be summarized as an attempt to identify face(s) detected in a probe image/video, using a gallery of stored faces.

From the above statement recognition can be seen to have two overall goals.

1. **Segmentation**: Detect faces from a still or video image(s)

2. **Identification**: Identify segmented faces within a stored database of faces

To achieve automatic face recognition, both the above goals must be achieved independently and unified to form our final recognition algorithm.

The segmentation component of the system, simply searches through the probe and attempts to segment all faces from it. Regardless of the type of probe, we initially use a trained detector to find potential faces. In the case of video, we take a further step by tracking the detected faces, in consequent frames.

Once all the faces are segmented, it is the role of the identification component to identify the subject. Identification is again not a single process or algorithm but a combination of components described below.
Figure 2.1: The components and flow making up a Recognition system for Still Image and Video Probes, based on diagram from [1]
Face Alignment is performed with the objective of achieving a more accurate localization and normalization of faces. Normalization can be done geometrically using properties such as size or pose, or using photometrical attributes such as illumination or gray scale. This is to ensure that faces we are comparing are considered under like conditions.

Faces in a recognition system, are stored as sequences of numbers (i.e a feature vector), which provide a unique encoding/signature for the individual.

Feature Extraction is performed to obtain the subject’s feature vector that is useful for distinguishing between faces of different persons.

Feature Matching compares pre-computed feature vectors of faces stored in our gallery with the feature vector extracted from the probe. The face is identified when the feature vectors are deemed to be similar (i.e. a comparison metric is below a specified threshold); otherwise it is unknown.

In video sequences, we are likely to have many varying faces for a given subject, over a number of frames, as opposed to a single image in still probes. The identity of the individual need not be found in every image we have, and therefore a framework is required to choose which of the face(s) should be identified (denoted as Framework For Multiple Faces in Figure 2.1).

2.2 Face Detection & Tracking

If we are unable to detect faces in the probe, we have nothing to identify, automatically resulting in a recognition rate of 0%. Hence the reliability of the detection component of the system has a direct influence on the performance of the entire face recognition system. Given any probe, an ideal segmentation system should be able to locate all faces present in the scene regardless of their position, scale, orientation, age or expression.

Face detection can be performed based on several cues: skin colour, motion (video specific), facial/head shape, facial appearance, or a combination of these parameters. Most successful face detection algorithms are appearance based without using other cues. The processing is done as follows: An input image is, scanned and classified as a face or non-face, at all possible locations and scales by a sub-window.

Segmentation in video is an extension of that in still images, where once the face is detected using the processing described above, it is then tracked in consequent frames.
Tracking systems can either take a model based or motion based approach. A motion based approach estimates the displacement of pixels from one frame to another. Whereas a model-based tracker uses a model of the object’s appearance and tries to change the object model’s pose (and possibly shape) parameters to fit the new frame. Although an imperative ingredient of a recognition system, the segmentation component does not lie at the centre of our investigation. Therefore an in-depth review of successful techniques in the area have been omitted. To achieve this functionality in the system, I utilise open source libraries for both detection and tracking. The theory relating to the basis of these libraries is provided in Chapter 3.

For the interested reader, Hjelmas & Low [2] provide a thorough Literature Review on Face Detection techniques. At the time of writing, no similar texts for face tracking could be found, however a good secondary source of information on the area can be found at [3]

2.3 Feature Extraction

The focus of this project is the identification algorithm, in which feature extraction plays a central role.

The high-level questions we are asking here are: What information of the face should be used to uniquely identify an individual? How do I obtain this information?

The face in machine vision terms is merely an array of pixel values. As a result of feature extraction, an input face will result in a 'signature' (i.e. feature vector) of the subject, which is later used for identification, using feature matching.

All faces share the same set of features (e.g. eyes, nose, mouth) arranged in the same configuration. The information that makes individual faces unique must be found in subtle variations in the form and configuration of the facial features. Early approaches to identification [4] took a literal approach to feature extraction, which relied on the geometry of fiducial points from the facial features (e.g. eyes, mouth corners etc) and their spatial relationships (e.g. distance between eyes etc).

This feature-based paradigm persisted for nearly 20 years, with researchers often disappointed by the low recognition rates achieved on even small data sets.
It wasn’t until the 1980s that researchers started experimenting with visual representations, making use of the appearance or texture of facial images. This paradigm shift from feature based to appearance based extraction led to the groundbreaking work of Kirby & Sirovich [5] with the Karhunen-Loeve Transform of faces in the 1990s. Based on this work, a family of subspace methods have been developed which have dominated the state-of-the-art in face recognition.

2.3.1 Feature Extraction in Subspaces

Images of faces, represented as high dimensional pixel arrays, often belong to a manifold of intrinsically low dimension.

As mentioned previously, we represent faces in our recognition system as feature vectors. In the image space, this feature vector can be obtained by simply writing the face image’s pixel values in a fixed (typically raster) order. Thus for an image of dimensions $m$-by-$n$ we obtain a feature vector in the $mn$-dimensional space.

A vector of this size even for modest images is high, and recognition suffers from a number of potential disadvantages (i.e. curse of dimensionality) including computational expense.

Moreover much of the surface of the face is smooth with a regular texture, and therefore per pixel sampling is in fact unnecessarily dense. In addition the appearance of the face is similar in overall configuration and is not randomly distributed in the image space (i.e. for a frontal face, the eyes fall on the side, nose in the middle etc).

On the whole not only does a high dimensional feature vector carry disadvantages for recognition, but much of the information it carries is redundant, as it does not enhance our ability to differentiate between faces. We can accurately describe a face in an intrinsically lower dimensional subspace (i.e. face space) of the image space.

This is shown in the following example. Consider a group of triplets (John, Paul, Bob). Being triplets they of course look the same for the most part, except: John has a mole above his lip, Paul has a mole below his lip, and Bob is lucky and does not have a mole. The main variation in this example is the mole. A recognition system just required to identify John, Paul and Bob would simply need to extract areas of the face above the lip and below the lip, as those areas vary the most.
Our lower dimensional feature vector can be imagined as 2 binary variables, for the mole above the lip and mole below the lip. Thus identifying a face has been reduced to identifying moles in 2 sub-areas of the face.

Although a rather contrived and crude example, it highlights that it is the variation in data that holds the key information that enables us to identify one face from another. When we look at a set of faces there are certain parts of the face that vary more than others, and this is the information we need to maintain to distinguish one face from another.

Subspace based feature extraction techniques analyse data in the image space, and identify correlation's between faces. These findings are then utilised to eradicate areas of high correlation, leaving a subspace of the image space where faces are represented in a more salient and relevant form than the original raw image.

Subspace techniques only vary by the manner in which they explore the training data to find forms of variation, from which they form the projection onto the face space. I shall proceed by outlining many of the techniques developed for this purpose.

**Principal Component Analysis (PCA)**

As a direct result of the work done by Kirby & Sirovich [5], PCA was utilised in the popularized 'Eigenfaces' technique of Turk & Pentland[6]. This is the original subspace feature extraction method, and is now classed as a de facto standard and used as a common performance benchmark in the field.

PCA analyses all images of subjects in a global space, and decorrelates the input data of the images using second-order statistics, (i.e. sample covariance).

Visually, PCA is finding a new set of axes (i.e. principle components) in the directions of maximum variation of the data set. The axes are directions specified by the eigenvectors of the covariance matrix of the original data, and hence the axes are orthogonal to each other as shown in Figure 2.2.

The first principle component accounts for as much variability in the data as possible, and each succeeding component accounts for as much remaining variability as possible.

This is formally stated as:
Figure 2.2: An example of PCA in action

The first principle component is the linear combination of the original dimension that has the maximum variance; the nth principal component is the linear combination with the highest variance, subject to being orthogonal to the n-1 first principle components.

Once the axes are determined, the first $n$ components representing the most variation in the data are selected as the axes for the face space. The feature vector for each face is then obtained by projecting the face from the image space onto the lower dimensional subspace made from the selected principle components.

Eigenfaces, is one of the algorithms which was implemented as part of the project and a more thorough treatment of this subject is provided in Chapter 4.

Independent Component Analysis

ICA is an extension of PCA, which removes the orthogonality constraint, and thus minimizes higher order dependencies as well, (as opposed to only second order dependencies in PCA) where the components found are designed to be non-Gaussian.

It is intimately related to the blind source separation, where the goal is to decompose an observed signal into a linear combination of unknown independent signals.

However it has been shown[7] that ICA based approach does not provide any advantage over PCA, in terms of accuracy and it computationally more expensive.
Linear Discriminant Analysis

Consider the case where our training set contains more than one image per subject, containing variation in illumination and expression.

Changes due to illumination for a given subject are always larger than between subjects of a fixed illumination. Clearly PCA would fall short at this hurdle, as the face space it would form would be based on the changes in illumination as opposed to identity.

As mentioned earlier, variation holds the key information for identification; however when we have more than one type of variation, we must cater for that when forming our subspace; otherwise we may extract irrelevant information, which will hinder our ability to differentiate between subjects.

Linear Discriminants Analysis (LDA)\[8\] forms a projection, which attempts to minimise intrapersonal variation, whilst maximising extrapersonal variation.

The technique initially forms a class for every subject in the training set. All images for a given subject are assigned to its respective class ($\pi_i$). It then forms two scatter matrices, the between-class and within-class, as shown below.

Let the between-class scatter matrix $S_b$ be defined as

$$S_b = \sum_{i=1}^{g} N_i (\bar{x}_i - \bar{x})(\bar{x}_i - \bar{x})^T$$

(2.1)

and the within-class scatter matrix $S_w$ be defined as

$$S_W = \sum_{i=1}^{g} (N_i - 1)S_i = \sum_{i=1}^{g} \sum_{j=1}^{N_i} (x_{i,j} - \bar{x}_i)^T$$

(2.2)

where $x_{i,j}$ is an $n$-dimensional data point $j$ from class $\pi_i$, $N_i$ is the number of training examples from class $\pi_i$, and $g$ is the total number of classes or groups.

The objective of LDA is then to find a projection matrix $P_{lda}$ that maximises the ratio of the determinant of $S_b$ to the determinant of $S_w$ (Fisher’s criterion), that is

$$P_{lda} = \arg \max P^T S_b P \left/ P^T S_w P \right.$$ 

(2.3)

Intuitively LDA tries to find the projection that maximises the variance of
the class means, whilst minimising the variance of the individual classes. Where PCA finds the directions for efficiently representing data, LDA seeks directions for discriminating data.

**A Bayesian Approach**

Variations between images of the same subject and those between different subject were considered separately in LDA. This idea is carried through in Bayesian Eigenfaces (BE)[9], and it can be viewed as a generalised nonlinear extension of LDA.

With LDA we formed a single subspace, where images of the same subject were clustered together and the distance between different subjects was maximised, enhancing our ability to discriminate between different subjects. In contrast, there are two key differences with Bayesian Eigenfaces:

- 2 independent subspaces are formed to characterise intrapersonal and extrapersonal variation separately
- The subspaces are spanned by the differences ($\Delta$) between two images, as opposed to the face images themselves

The difference image for two face images is simply the signed arithmetic difference between respective pixels in the source images. Such difference images fall into two distinct classes, either intrapersonal or extrapersonal; both modelled as approximately Gaussian.

Their final classifier matches probe images to stored images by computing the likelihood that the corresponding difference images came from the subspace of the intrapersonal ($P(\Delta|\Omega_I)$) rather than extrapersonal ($P(\Delta|\Omega_E)$) difference images.

PCA, is applied twice to form a compact representation of both subspaces, leaving the most significant dimensions. After training, for each subspace we have:

- A projection matrix $\phi_M$ and a vector of eigenvalues $\lambda$
- A parameter $\rho$ that is the average of the eigenvalues between $M$ and $T$ (where $M$ is the number of dimensions maintained and $T$ is the number of difference training images)
- The projection, $y = [y_1y_2\ldots y_M]^T$, of each difference image $\Delta$ into the PCA subspace
PCA simplifies the analysis by accomplishing two things. First, it provides a linear transformation matrix $\Phi$ that rotates the training data in such a way that the variances along each dimension are uncorrelated. Second it provides estimates for the variances along those dimensions. In short, PCA provides all the parameters needed to estimate the probability function for a Gaussian distribution given a set of samples:

$$
\hat{P}(\Delta|\Omega) \equiv \frac{\exp\left(\frac{-1}{2} \sum_{i=1}^{M} \frac{y_i^2}{\sigma_i^2}\right)}{(2\pi)^{M/2} \prod_{i=1}^{M} \sigma_i^{2}}
$$

(2.4)

where $y = \Phi^T \cdot \Delta$ and is an $N$-dimensional vector and $\sigma^2 = \lambda$.

Non-linear Subspace Techniques

So far the techniques we have outlined, have formed subspaces based on a linear principal manifold.

Due to the complex nature of faces, assuming a linear principal manifold is highly unlikely. Non-linear methods[8] offer greater flexibility to the underlying manifold and extend our ability to detect and eradicate higher order correlations, thus improving our ability to discriminate between faces.

In previous techniques the projection onto the face space amounted to a projection matrix. Nonlinear projections require a more complex operation to project a face from the image space onto the face space.

One of the simplest methods for computing non-linear principal manifolds is the Non linear PCA (NLPCA) autoencoder multilayer neural networks, which produces a nonlinear projection function conveyed as a weighted sum of sigmoids in the neural network.

2.3.2 Hybrid Approaches

Another approach to handle the nonlinearity of faces in the image space, is to construct a local appearance-based feature space, using appropriate image filters, so the distributions of faces are less affected by various changes.

Algorithms of this form may be considered as combining geometric (or structural) feature detection and local appearance feature extraction, to increase the stability of recognition performance under changes in viewpoint, illumination, and expression.
Elastic Graph Bunch Matching (EGBM)

EGBM is the common method adopting this approach, and was developed by Wiskott et al.\cite{10} as an extension to their original graph matching system, in which faces are represented as graphs with nodes.

The algorithm initially finds landmark locations on the images that correspond to facial features such as eyes, nose etc. It then uses Gabor wavelet convolutions at these points to describe the features of the landmark.

A face graph is then created by placing each node at the landmark locations, and the extracted Gabor jet describing the feature is stored at the node. The similarity of two images is a function of the corresponding face graphs.

2.4 Invariant Recognition to Pose Variation

Many face recognition systems have shown a considerable drop in performance when dealing with variations in pose. The problem arises due to the stark difference in the configuration of faces from various viewpoints.

Unlike constraints such as illumination or background, the pose of an individual is something we cannot attempt to restrict. In videos we have less control over a subject’s pose, thus making it an extremely important and worthwhile problem to address.

Researchers have proposed various approaches to deal with variations in pose: (1) Multi-Image (2) Hybrid (3) Single Image.
2.4.1 Multi-Image Based Approaches

Multi-Image based techniques adopt a brute force approach to pose invariant recognition, where multiple images of every subject are required at every pose we wish to identify them at. It can be compared to the notion of overfitting in machine learning, where no generalisations are made from the training data.

As we are effectively overfitting, the main limitation with this approach is that we cannot effectively recognise novel poses, which are not present in the gallery. Consequently methods of this approach suffer from computational cost and lack of scalability.

Identification is performed using techniques characteristic of the feature based paradigm, where features (e.g. eyes, mouth etc.) are extracted from the subject, and a feature vector is formed on the basis of spatial relationships between the extracted features.

Beymer et al.[11], utilise this approach in which the eyes, nose and mouth of the subject, are extracted using a template. These features are then used to estimate the pose of the face, which is stored along with its feature vector.

When it comes to identification, the previously estimated pose for gallery images is utilised and images are only compared if they are believed to be in a nearby pose to that of the probe, thus avoiding exhaustive matching with all gallery images.

2.4.2 Hybrid Approaches

To date, hybrid approach methods have offered the most successful and practical methods, to pose invariance. Multiple images of a subject are utilised at the training phase, but are not required in the gallery unlike Multi-Image based approaches.

Pentland et al. [12] present an extension the popular eigenfaces approach to feature extraction, with view based and parametric eigenspaces.

View-based eigenspaces creates a separate subspace for each viewpoint present in our gallery. In order to choose which eigenspace to perform identification in, pose estimation is required beforehand.

Figure 2.4 shows the advantage of a View Based architecture. Each subspace formed for View Based Eigenfaces characterises images of different subjects, under the same pose. Therefore the variation extracted by the Principal
Components relate to that between different subjects, and not between variations in pose of the same subject. This is shown by the fact that different faces of the same subject project to the same region in the subspace, enabling us easily classify a new face based on its distance from the clusters.

However with Parametric Eigenspaces (i.e. a single global eigenspace), all images are projected onto the same subspace. Therefore, the variation extracted is not necessarily that between the different subjects. As shown in Figure 2.4, images of the same subject are not necessarily projected to the same area of the subspace. Due to this, classification is no longer as clear cut, and can be error prone.

Figure 2.4: Plotting the first two principle components of images with varying pose, using View-based, and Parametric Eigenspaces

2.4.3 Single Image Based Approaches

The final approach attempts to create a pose invariant image representation of the face from a single image.

As discussed previously, a Gabor [13]wavelet image representation offers limited success and is robust to small angle rotations, when used in Elastic Graph Bunch Matching.

A popular representation has been to use 3D face models, which are formed by fitting a single image of a face to a polygonal or mesh model which simulates tissue. The model is then used to generate the subject at any given pose, as detected in the probe. Once a 3D model is formed, the illumination invariance problem can also be solved with minimal additional work.

The downside of generating a 3D model is its complexity and computational expense, which have restricted its success and applicability.
Everingham & Zisserman [14] utilize a single image based approach, with a model based algorithm. They attempt to form a 3D model of an individual using coarse 3D geometry with multiple texture maps. Using this model they are able to extrapolate some way from a single view of the person and propose how the person looks in nearby poses.

Classification of a face initially requires the pose of the probe to be estimated. Once acquired each individual present in the gallery is recreated at that pose, using its 3D model. Multiple appearances may be used, to generate the individual with varying expression or illumination. Descriptors for the probe and each model are generated using image gradients, which are compared and classified.

Experiments were conducted by attempting to identify the 3 main characters for an episode of the sitcom Fawlty Towers. Characters were identified in 75-95% of frames at a false positive rate of 10%.

2.5 Invariant Recognition to Illumination Variation

Appearance-based methods of feature extraction have proved to be extremely successful in Face Recognition Technology. However as these methods use image intensities directly they are inherently sensitive to variations in illumination.

Drastic changes in illumination therefore cause significant problems for appearance-based recognition algorithms, making it an important issue, which must be addressed to ensure robust recognition.

Many approaches to resolve this problem have been suggested: (1) Class-Based (2) Heuristic Based (3) Model Based (4) Image Comparison.

2.5.1 Class Based Approaches

Techniques using a class based approach arise from the underlying assumption formed from Lambertian surfaces. That is the luminance of a Lambertian surface is constant regardless from which angle it is viewed from. Using this premise we can infer that there exists a 3D linear subspace within our image space which provides a illumination invariant signature of a given subject.

Class Based techniques are similar to dimensionality reduction methods such as PCA, in that they produce a dimension lowering linear projection. In this
case however, we are attempting to identify the 3D subspace for a given person. Therefore we require a class of images of a person, consisting of the subject under different lighting conditions from a fixed pose.

Fisher Faces [15], was presented earlier and is an example of a class based method which attempts to shape the scatter produced by the linear projection in order to make it more reliable for classification. Its objective is to minimise the within-class scatter, whilst maximising the between-class scatter\(^1\). Experiments carried out with 5 people under drastic lighting variation, showed fisher faces achieved a significantly lower error rate than eigenfaces.

### 2.5.2 Heuristic Approaches

Algorithms utilizing a heuristic approach, simply provide a loose set of rules to achieve its objective. One such technique[15] suggests that discarding the three most significant principal components when using eigenspaces, reduces variations due to lighting.

Unfortunately to achieve illumination invariance rigorously such simplistic rule based methods have been found thus far to be inadequate when tested extensively.

### 2.5.3 Image Comparison Approaches

We can try and reduce illumination by modifying the way in which we represent our image. With a image comparison approach, images are transformed into a representation minimizing variations in illumination.

For example a technique which epitomizes the objective of such an approach would be to simply transform all gallery and probe images into edge maps. Edges are usually determined by comparing its relative intensity with neighboring pixels. As the 3D structure of a face remains constant the ratio of a pixel’s intensity compared to its neighbours will remain fairly constant under variations in lighting. Therefore a person in a fixed pose and subject to large variations in illumination would result in the same edge map.

Gross et al. [16] adopt a similar approach producing a preprocessing algorithm using Bayesian face subregions, removing illumination variation via normalization.

The approach uses two widely accepted assumptions about human vision; in that it is:

\(^1\)A class would consist of various face images of a given individual under varied lighting
1. mostly sensitive to scene reflectance and mostly insensitive to the illumination conditions.

2. responsive to local changes in contrast rather than to global brightness levels.

In general the intensity \( I(x, y) \) of a pixel in an image is regarded as the product of the reflectance \( R(x, y) \) and illuminance \( L(x, y) \) at the point.

Their method attempts to estimate the \( L(x, y) \), such that when it divides \( I(x, y) \) it produces \( R(x, y) \) in which the local contrast is appropriately enhanced.

### 2.5.4 Model Based Approaches

The aim with a model based approach is to form a 3D model of a subject from 2D images. We create our model based on the pixel intensities in a 2D image, using methods termed as shape-from-shading (SFS) algorithms. We can then produce the subject at any illumination by simulating a light source onto the generated model. Light source estimation for the probe is required for a fully automated model based approach.

Although model-based techniques have shown significant performance improvements over other techniques, they are computationally expensive and are not scalable.

### 2.6 Selection of Frames for Recognition

The application of face recognition to video is still very much in its infancy, and has only received attention in the last 5 years.

"Tracking then Recognition" was the first approach developed to facilitate the application of face recognition techniques to videos. With this approach a detected face is tracked until a frame satisfying certain criteria (e.g. pose, size) is acquired. Recognition is then attempted with the chosen frame using still-to-still recognition.

This simple heuristic approach to frame selection does not take advantage of the probe face being present in many frames. Moreover it does not effectively exploit temporal information available in videos.

More recent approaches have proposed probabilistic frameworks, which fuses temporal information and utilize more than one frame the face is tracked in as evidence towards recognition.
Chellappa and Zhou [17] propose a "Tracking and Recognition" approach that attempts to resolve uncertainties in tracking and recognition simultaneously in a unified probabilistic framework. Temporal information is fused using the time series state space model, to characterize the evolving kinematics and identity in the probe video.

The framework presented comprises of 3 components:

- A motion equation governing the kinematics of the tracking motion vector.
- An identity equation governing the temporal evolution of the identity variable.
- An observation equation establishing a link between the motion vector and the identity variable.

This framework can take advantage of any still-to-still recognition algorithm by embedding distance measures into the likelihood measurement.

The framework was tested using various still-to-still recognition algorithms. Notably, 83% of faces were correctly identified when the framework was integrated with a probabilistic subspace measure for the likelihood measurement of the identity.
Chapter 3

Face Detection & Tracking

The first task of any recognition system is to process the probe and segment all the faces that appear in it. In this chapter, I walkthrough the technology used to provide this functionality in *enVisage*.

3.1 Detection

Regardless of whether we have an image or video probe, segmentation initially requires the ability to detect faces. Intel’s Open Source Computer Vision Library (OpenCV) [18] is used to provide this functionality, from which we are able to form a multi-view face detector that is able to detect faces with out of plane rotation ranging from -90 to +90 degrees.

3.1.1 A Statistical Based Classifier

Open CV provides a statistical based classifier as part of its library, which can be trained to detect any arbitrary object.

Training requires both positive (e.g. faces) and negative (e.g. non-faces) examples of the object of interest. During training, distinctive features are extracted from the training samples which are later used to classify the object in unseen examples. Information obtained whilst training is compressed into a statistical model. However if the trained detector misclassifies new examples, the model parameters can be amended accordingly.

The detector provided is based on the approach developed by Viola and Jones[19], which offers robust real-time object detection.

3.1.2 Robust Real-time Object Detection

Viola & Jones [19] describe a visual object detection framework, that is capable of processing images extremely rapidly while achieving high detection
rates. This method uses simple Haar-like features (so called because they are computed in a similar method as the coefficients in Haar wavelet transforms) and a cascade of boosted tree classifiers as a statistical model.

**What is a Feature?**

Each distinctive feature identified from the training set, is described by a template (shape of the feature), its coordinate relative to the search window origin and the size (scale factor) of the feature. A template consists of two or three joined, black and white rectangles, either up-right or rotated by 45°.

![Haar Features](image)

The Haar feature’s value is then calculated as a weighted sum of two components: The pixel sum over the black rectangle and the sum over the whole feature area (all black and white rectangles). The weights of these two components are of opposite signs, and for normalization their absolute values are inversely proportional to the areas.

The computed feature value is then used as input to a very simple decision tree classifier that usually has just two terminal nodes, that is:

\[
 f_i = \begin{cases} 
 +1, & \text{if } x_i \geq t_i \\
 -1, & \text{if } x_i < t_i 
\end{cases}
\]

where the response +1 means the face, and -1 means the non-face. For example, in many face images eyes are darker than the surrounding regions, and so feature 3a in Figure 3.1, centered at one of the eyes and properly scaled, would give a large response (assuming that weight_{black}<0).

A single feature (weak classifier) will relate to a certain feature or characteristic of the face, and is not solely able to detect a face. In real classifiers, hundreds of features are used. Direct computation of pixel sums over multiple
small rectangles would make the detection very slow. To reduce computation
an elegant method to compute the sums quickly using a Integral Image, is
introduced. The Integral Image which is a Summed Area Table (SAT) is
computed over the whole image.

\[
SAT(x, y) = \sum_{x, y} I(x, y).
\]

![Integral Image Representation](image)

The pixel sum over a rectangle can then be computed from the corners of a
rectangle with the Integral Image, as shown in Figure 3.2. Thus the area of
a rectangle is reduced to 4 accesses of the SAT as opposed to the \((\text{rect}_\text{width} \times \text{rect}_\text{height})\) pixel access.

**Cascading Classifiers**

Using the weak classifiers, we then combine their outputs as a weighted sum
as follows to determine if we have a face or not:

\[
F = c_1 f_1 + \ldots + c_n f_n
\]  

(3.1)

Where the weights \(c_i \ldots c_n\) are based on the error evaluated in capturing \(f_i\)
from the training set. The smaller the error in calculating \(f_i\) on the training
set, the larger the coefficient \(c_i\). This way greater importance is given to
features which consistently appear on the positive examples of our training
set, and don’t on negative examples.

As opposed to a single classifier where all features are evaluated at once,
Viola & Jones suggest using a cascade of classifiers.

The classifiers are assigned a subset of features increasing in complexity,
where the initial classifier is assigned the most occurring features with high
\(c_i\), and the last classifier is assigned the least occurring features. With this
method, non-face windows are identified quickly without any unnecessary
time being spent in calculating other features.
3.1.3 A Multi-view Detector

Intel’s library provides training data for both frontal and left profile faces. The face detector implemented utilises both data sets, and attempts to detect frontal, as well as left and right profile faces. We serially search the image for frontal followed by left profile, and finally right profile faces. To detect right profile faces the image is flipped in the horizontal axis before being searched by the left profile detector. The $x$-coordinates of faces objects detected are appropriately altered for the original image.

Merging Faces

As the probe is passed through three detectors, multiple detections of a given face can occur. To avoid multiple segmentations of the same face, a merging algorithm combines detected faces, if they are in close proximity. This algorithm assumes that occluded faces are not detected.

Proximity of faces is determined by the distance between the top left coordinates of nearby faces. If the Euclidean distance between the points is below a threshold ($\tau$), both faces are removed and an average of both removed faces is inserted, as a new face.

Faces are stored in order of their $x$ coordinate of the top left corner of the
face. Faces are compared to neighbouring faces if the difference in the x coordinate is itself below $\tau$. This is done in both the positive and negative x direction (i.e. higher and lower array indices from a given location).

![Detection without merging](image1.png) ![Detection with merging](image2.png)

Figure 3.5: *An example of the merging algorithm in action*

### 3.2 Tracking

In video sequences, a subject appears over a number of frames, from which the faces must be segmented. Once detected using the the face detector described above, the face is then tracked in consequent frames using a colour based tracker. We use CMU’s [20] widely used tracking library, which is combined with the aforementioned face detector to segment faces in video probes.

Another option which was considered was to repeatedly use the face detector over all frames, and cluster the results. This approach however limits the movement over which we can track the head, to out of plane rotation and translation, which is not the case with a colour based tracker.

#### 3.2.1 Stochastic Colour Based Tracking

The face tracking system is based on statistical colour modeling[20] and a deformable template. Essentially the tracker modifies the face region, from one frame to the next, to maintain as many face coloured pixels as possible.

Given that a face has been detected at frame $n$, the tracker initially trains itself, by determining what face, and non face coloured pixels are. This is done by evaluating the probability of a given colour (for all possible colours a pixel can take) belonging to each of the two classes: the 'face' class and the 'non-face' class from the detected frame. The face class is made up of all pixels inside the detected face area, and the non-face class is all pixels which lie outside it.
Gaussian Mixture models are used to approximate the colour distributions of the two classes. It is assumed that there are two dominant colours in the face, the skin and facial hair colour. Therefore our Gaussian Mixture model is a weighted sum of two Gaussians. This assumption of two dominant colours is carried through for the non-face region, as well.

The probabilistic model is then utilised to deform the detected face (i.e. the template) in consequent frames such that the template includes as many face pixels as possible whilst excluding as many non-face pixels as possible. This amounts to an optimization problem, where the objective is to identify a region \( R \) in the new frame such that the below equation is minimised.

\[
f(R) = \min \sum_{\forall \text{pixel} \in R} \log \left( \frac{P(p_{\text{colour}}|\text{Not Face})}{P(p_{\text{colour}}|\text{Face})} \right)
\]

3.3 Unifying Detection & Tracking

The manner in which we unify the tracking and detection modules directly impacts the system performance, in terms of the number of faces correctly segmented as well as computational expense.

A simplistic model developed initially, was where detection and tracking were performed serially. With this approach we are either attempting to detect faces, or tracking previously detected faces, but never both. Therefore if a new face appears in one of the frames where we are tracking a previously detected face, it would be missed. Clearly this approach is extremely limiting, as many faces would go undetected.

This approach was extended where multiple faces appearing at any time could be detected and tracked.

With this modified approach, the detector is applied to every frame of the video sequence, hence all faces are detected. Any faces which are detected creates a instance of the tracker in a new thread with the responsibility of only tracking the detected face in consequent frames.

However the detector could possibly performing unnecessary work, as it may detect a face which is currently being tracked. To avoid this scenario, a frame is initially processed by all trackers which may exist. The detector is then applied to the frame, but is constrained such that it does not search for new faces in the regions annotated by the trackers. The above models are shown in Figure 3.6
Figure 3.6: The evolution of unifying detection and tracking, from a serial to concurrent model.
3.4 Manual Annotation

As detection and tracking were not the focus of this project, manual means of annotation of a video or image probe are provided via the UI. The functionality provided includes, segmenting faces by ’drawing’ on the image, tracking and updating a detected face in neighbouring frames, as well as providing information regarding the pose of the segmented face.

The purpose of implementing this additional functionality is to ensure that we provide the necessary framework to facilitate and test the identification algorithm in case our automatic tracking and detection system fails.
Chapter 4

Design of a Novel Algorithm for Identification

The identification algorithm of our system lies at the heart of this project. It was our aim to integrate a number of techniques in research to form an algorithm which is invariant to changes in pose and illumination.

In this chapter I will outline the exact aims of the algorithm, and provide a high level justification for the techniques to be utilised to achieve them. Chapter 4 contains specific details on how this algorithm was implemented.

Figure 4.1: An Overview of the components making up an Identification Algorithm

4.1 The Problem

The algorithm, as stated previously, must be invariant to changes in pose and illumination. Specifically, our algorithm should be able to identify individuals which exhibit:

- Out-of-plane variation ranging from -90 to +90 degrees
• Wide range of illumination, varying in location and intensity of light source

Figure 4.2: The algorithm should be able to identify the subject with any of the above forms of variation, which may occur separately or in unison

4.2 The Solution

4.2.1 Basis of the Algorithm

In Chapter 2, we outlined the different forms of algorithms which identification can be based on; Feature-Based, Hybrid, and Appearance Based.

Feature based algorithms, were the first attempt to recognise faces, using features of the face and the spatial relationships between them. Although feature extraction methods originated from the feature based paradigm, the performance of feature based techniques has been extremely poor, and have generally been abandoned since the introduction of appearance based methods.

Hybrid approaches have proven to be extremely successful, characterised by the performance of Elastic Graph Bunch Matching. However these techniques are reliant on salient facial feature points which are difficult to automatically extract robustly. In addition, feature matching is often a computationally intensive process, requiring relatively high resolution and good quality face images.

Appearance Based methods, have been the most successful to date. They are computationally inexpensive in comparison to other methods, requiring no additional annotation, and are applicable to low resolution images. For this reason, we form our algorithm using the appearance based paradigm.
4.2.2 Observations

The algorithm was formed from a number of "appearance-based" observations from the types of variations exhibited in Figure 4.2.

1. Large differences in profile and frontal views

As seen above, faces vary considerably in configuration from profile ($\pm 90^\circ$) to frontal ($0^\circ$) views. The two faces would project to far apart points in the face space, even though they belong to the same subject. As mentioned previously it is variation that holds the important information for identification. However the variation we require is that between individuals and not large differences between images of the same subject.

We adopt a view-based approach where we split the identification problem into two, forming two independent subspaces for profile and frontal views.

There is an added overhead with the detection stage with this approach, in that the pose of the face will have to be extracted. This is later used to determine which subspace is used to identify the probe. In most cases we have separate detectors trained for frontal faces and profile faces respectively, making this information readily available.

2. Small Variations in Pose

Poses in between the frontal and profile views also exhibit variation, but to a lesser extent. To account for this form of variation, we use Bayesian Eigenfaces which effectively characterises subtle interpersonal variations.

To characterise this variation, our training set will require images of a subject exhibiting these subtle changes in pose.

In addition, we geometrically normalise all near frontal faces, using annotation of the eyes (where both eyes are detected), which slightly reduces the variation.

3. Symmetry in Pose (non-frontal)

We note that right profile images are near mirror images of left profile images. Therefore it is not necessary to maintain a subspace for each. We simply form a subspace for left profile images. If a probe detected as a right profile, we flip the face and project it onto our left profile subspace for identification.

4. Differences in Illumination

All images (probes and gallery images) are subject to photometric normalisation, thus reducing variation in
Figure 4.3: Small Variations in pose, can be reduced by performing Geometric Normalisation, using the annotation of the eyes (*images have been generated using enVisage)

Figure 4.4: Symmetry which occurs between right profile and left profile faces
illumination. We utilise Gross and Brajovic’s algorithm for this purpose.

![Image of effects of photometric normalisation using Gross & Brajovic’s algorithm](image)

Figure 4.5: Effects of photometric normalisation using Gross & Brajovic’s algorithm (*images have been generated using enVisage*)

### 4.3 Summary

From the observations above, the technology required to be developed for our algorithm is:

- Eigenfaces[6]
- Bayesian Eigenfaces[9]
- View Based Eigenfaces[12]
- Eye annotation and Geometric Normalisation
- Gross & Brajovic[16] Illumination Normalisation algorithm
Figure 4.6: A flowchart representation of the Novel Algorithm for Identification against Pose and Illumination
Chapter 5

Implementation of a Novel Algorithm for Identification

In the previous chapter, we outlined a blueprint for a novel identification algorithm. In the following chapter, I provide technical details for each component of the algorithm relating to its implementation.

5.1 Eigenfaces

As discussed in the background, Eigenfaces is a subspace feature extraction technique utilising Principal Components Analysis. In this section, I provide an in-depth analysis of the implementation of Eigenfaces.

5.1.1 An Overview

To recap, Eigenfaces\cite{6} is a linear subspace technique. The aim is to form a lower dimensional subspace of the image space, where only the data exhibiting the most variance is maintained. We want variation, as that is what enables us to distinguish from one individual to the next. Eigenfaces has two phases:

- A training phase, in which a set of faces are explored to ascertain areas of the data exhibiting the most variance. The result of which, is a projection onto this lower dimensional area

- A testing phase, where faces are compared after being projected onto the face space extracted from the training phase

The Training Phase

The faces in the training set, are explored using sample covariance ($2^{nd}$ order statistic) as a metric for the variance. The axes of the lower dimensional
subspace are orthogonal to each other, which is a constraint of PCA. The projection matrix $T$, onto the face space, is formed as a result of training.

**Algorithm 1** To obtain the face subspace (Training Phase)

1. A set of Training Images $P \leftarrow \{p_1...p_w\}$, where $p_i \in \mathbb{R}^{m \times n}$ (where $p_i$ is a face image)
2. Initialise Matrix $A \in \mathbb{R}^{(m \times n) \times w}$ (All images in training set are later written in $A$)
3. Evaluate Mean Image, and form $\mu \in \mathbb{R}^{m \times n \times 1}$ by reading its pixel values (raster order)
4. for all $p_i \in P$ do
5. Read $p_i$ (raster order) forming column vector $c_i \in \mathbb{R}^{m \times n \times 1}$
6. Set $i^{th}$ column of $A$ to $c_i - \mu$
7. end for
8. Evaluate Covariance Matrix $D$ of $A$ ( $D \leftarrow AA^T$ )
9. Perform Eigenanalysis on $D$
10. Select top $k$ eigenvectors (with highest eigenvalues)
11. Form projection matrix $T \in \mathbb{R}^{(m \times n) \times k}$, where columns are the selected Eigenvectors (This is the projection matrix used to project images into the face space, for testing)

### 5.1.2 Testing Phase

Once the face space has been evaluated through the training phase, all gallery images are projected onto it (using the projection matrix $T$). The resulting lower dimension feature vectors are stored.

For identification, a probe is also subject to the same projection($T$) and its feature vector is compared with every stored feature vector, using a Euclidean distance.

The stored vector yielding the lowest distance to the probe’s feature vector is the identity of the probe, if it is below a specified threshold.

### 5.1.3 Reducing the Cost of Training

The method outlined above would lead to a extremely large covariance matrix $D$. For example, for 10 training images of size $64 \times 64$, we would have a covariance matrix of $4096 \times 4096$. Not only does the covariance matrix of this size become computationally expensive to evaluate, but determining the eigenvectors / eigenvalues of such a large covariance becomes an intractable task.
Figure 5.1: A flowchart outlining the Testing phase of Eigenfaces
It is known that for a matrix of $M \times N$, the maximum number of useful eigenvectors is the $\min(M - 1, N - 1)$. Since the number of training images ($N$) is usually considerably less than the number of pixels ($M$), the most eigenvectors/eigenvalues that can be found are $N - 1$.

A common theorem in linear algebra states that the eigenvalues of $AA^T$ and $A^T A$ are the same. Furthermore, the eigenvectors of $AA^T$ are the same as the eigenvectors of $A^T A$ multiplied by the matrix $A$ and normalized.

Thus once we have evaluated $A$ as shown above we can drastically reduce computation[6] of the projection matrix by the following changes:

**Algorithm 2** Modification to Training Phase

Evaluate Covariance Matrix $D$ of $A$ ( $D \Leftarrow A^T A$ ) {Lines 1 .. 7 remain the same}
Perform Eigenanalysis on $D$
Select top $k$ eigenvectors

**for all** Selected eigenvector $e_i$ $\Leftarrow 0..k$ **do**
  Multiply $e_i$ by the original data matrix $A$
  Normalize $e$
**end for**

Form projection matrix $T \in \mathbb{R}^{(m \times n) \times k}$, where columns are the selected revised Eigenvectors {The projection matrix, $T$ obtained is the same projection matrix as before}

The resulting projection matrix $T$ is the same as obtained previously, and the testing stage of the algorithm remains the same as outlined in Figure 5.1.

### 5.1.4 How many Eigenvectors?

An important and largely unsolved problem in dimensionality reduction is the choice of "top $n$ eigenvectors" which determines the intrinsic dimensionality of the principal manifold. The problem lies in identifying which Eigenvectors correspond to useful information and which are simply meaningless variation.

By looking at the images of specific Eigenvectors, it is sometimes possible to determine what features are encoded in that Eigenvector. Eigenvectors 1 and 2 seem to encode the illumination in the image from left to right, and top to bottom respectively. Eigenvector 5 shows a faint outline of glasses. The higher order Eigenvectors (100, 150, 200) become blotchy and it becomes difficult to identify what they are encoding.
Given that Eigenvectors represent the amount of variance along a particular Eigenvector, traditional approaches have opted to select the Eigenvectors with the largest Eigenvectors (i.e. dimensions along which our training set of images vary the most). Thus the last Eigenvectors with lower values Eigenvectors were eliminated.

There are three variations of deciding how many of the last Eigenvectors to eliminate:

1. Remove last 40% of Eigenvectors (a heuristic formed from my experience).

2. A Minimum number of Eigenvectors which guarantee an energy $\varepsilon$ above a threshold (typically 0.9) where Energy $e_i$ of $i^{th}$ eigenvector is

$$ e_i = \frac{\sum_{1 \leq j \leq i} \lambda_j}{\sum_{1 \leq j \leq k} \lambda_j}$$

(5.1)

3. All eigenvectors with a stretch $s$ greater than a threshold (typically 0.01) are retained where stretch $s_i$ of $i^{th}$ eigenvector is

$$ s_i = \frac{\lambda_i}{\lambda_{\text{largest}}}$$

(5.2)

Comparison between different values of energy and stretching was conducted as described below:

- **Dataset** of 20 subjects with 3 variations in small changes in expression (none, smiling, eyes shut)

- **Gallery** consists of 1 image per subject with no expression

- **Training** performed using all images in gallery

- **Testing** performed on remaining 2 images per subject

As shown above, optimal recognition is achieved with an energy threshold of approximately 0.9, or a stretch of 0.01. We utilise energy with a threshold of 0.9 for our system.
5.2 Bayesian Eigenfaces

Using Eigenfaces, as described above, we are characterising faces by exploring the training images.

With Bayesian Eigenfaces[9] we explore, the differences between images of the same subject, as well as differences between images of different subjects, and not the images themselves.

We are no longer characterising faces in a single subspace, but are using PCA twice to form two independent subspaces to characterize both intrapersonal (differences between images of the same subject) and extrapersonal variation (differences between images of different subjects).

Description of Algorithm

The Training phase of the algorithm performs PCA (as outlined previously) twice, on two sets of training images.

The training set for the intrapersonal subspace, is made up of the difference ($\Delta = I_1 - I_2$) between pairs of images $(I_1, I_2)$ of the same subject under varying conditions (lighting, expression, illumination).

Similarly, the training set for the extrapersonal subspace, is made up of the
difference image between pairs of images of different subjects.

PCA is performed separately for both training sets, resulting in two projection matrices onto the intra / extra personal subspaces.

Computing the similarity score involves initially subtracting a candidate image \( I_j \) from a database entry \( I_k \). The resulting \( \Delta \) is then projected onto both subspaces. The exponentials are then evaluated, normalized and combined into likelihoods as shown below:

\[
S(I_1, I_2) = \frac{P(\Delta|\Omega_I)P(\Omega_I)}{P(\Delta|\Omega_I)P(\Omega_I) + P(\Delta|\Omega_E)P(\Omega_E)}
\]  
(5.3)

This operation is iterated over all members of the database. The image resulting in the maximum score is found as the match.

### 5.2.1 Efficient Similarity Comparison

The above evaluation becomes extremely expensive with large database. However these computations can be greatly simplified by offline transformations.

\[
i_j = \Lambda_I^{-1/2}V_I I_j
\]  
(5.4)

\[
e_j = \Lambda_E^{-1/2}V_E E_j
\]  
(5.5)

where, \( \Lambda \) and \( V \) are the matrices of the largest eigenvalues and eigenvectors \( \Sigma_E \) or \( \Sigma_I \).

To compute \( P(\Delta|\Omega_I) \) and \( P(\Delta|\Omega_E) \) we pre-process the \( I_k \) images with whitening transformations, for both intrapersonal and extrapersonal subspaces as shown in equations 5.4 & 5.5 respectively. Consequently every image is stored as two vectors \( (i_j, e_j) \) of whitened subspace coefficients.

The whitening transforms \( \Lambda^{-1/2}V \) transform the \( I \) image difference vector, such that the resultant vector (i.e whitened subspace coefficient) is white (i.e. its components are uncorrelated and their variances equal unity. In other words the covariance matrix of the whitened subspace coefficients is the Identity matrix.

After preprocessing, evaluating the likelihoods is reduced to computing simple Euclidean distances for the exponents.
In this manner, one avoids unnecessary and repeated image differencing and online projections. The MAP similarity measure can now be evaluated by substituting the above values. The \( P(\Omega) \) terms of the equation can be set to reflect specific operating conditions (e.g. number of test images vs. the size of the database). An alternative probabilistic similarity measure (ML) can be defined in simpler form using the intrapersonal likelihood alone,

\[
P(\Delta|\Omega_I) = \frac{e^{-1/2||\epsilon_j-\epsilon_k||^2}}{(2\pi)^{D/2}||\Sigma_E||^{1/2}}
\]

\[
P(\Delta|\Omega_I) = \frac{e^{-1/2||l_j-l_k||^2}}{(2\pi)^{D/2}||\Sigma_I||^{1/2}}
\]

(5.7)

Moghaddam & Pentland[9] report only a slight performance gain of ML over MAP of approximately 2-3%. At the same time the computational cost of the ML measure is half that of the MAP. For that reason we utilise the ML measure in our system.

The normalizing denominators of both these equations could be pre-computed offline. However, the terms of the denominator are all additive constants and do not affect the relative probabilities. For classification the unnormalized value is suitable, so we need not compute the denominator in any of the above equations, increasing performance for training as well as testing. In addition, to bring the distance measures in a reasonable range we take the log likelihood of the value. This was not presented in Moghaddam & Pentland’s paper.

5.2.2 A Computational Shortcut for the Covariance

Computation of the covariance is considerably reduced, using the Pooled Covariance Estimate (PCE). The PCE has been successfully utilised in Linear Discriminant Analysis. With this approach we avoid calculating the covariance of all individuals explicitly, by computing the covariance as a weighted sum of each individual’s own covariance matrix. The final covariance is approximately the same, whilst computational costs are drastically reduced.

5.3 View Based Eigenfaces

The extension to view based eigenspace[12] from Eigenfaces is a simple architectural extension which enables the supports multiple instances of training data for separate subspaces. We form two independent Eigenspaces for
**Algorithm 3** Calculating PCE

1: \( N \) individuals with \( M \) unique difference images \((I_1 - I_2)\)

2: Form matrix \( X_i \) \( \{i = 0 \ldots N\} \) for each individual where, \( j^{th} \) row is rasterized image of person \( i \), for \( j = \{1 \ldots M\} \)

3: Initialise \( \Sigma_I = 0 \)

4: for \( (i = 0 \ldots N) \) do

5: \( \Sigma_I = \Sigma_I + \text{Covariance}(X_i) / M \)

6: end for

7: \( \Sigma_I = \Sigma_I / N \)

When images are added to the gallery, they are manually annotated with pose via the UI. When the system is trained, we perform the training phase of the feature extraction algorithm twice for frontal and then left profile gallery images (where right-profile images are flipped before training), forming two independent subspaces. The training algorithm utilises images from the gallery which are annotated with the appropriate pose for each subspace (i.e., only frontal images used to form frontal subspace).

For the testing phase, the algorithm uses the pose of the probe and assigns it to the appropriate subspace. The probe is either annotated as "Frontal", "Left-Profile", or "Right-Profile". This information is obtained and maintained at detection, depending on which detector detected the face in question. Alternatively, this information can be manually entered via the UI. A face which is detected by multiple detectors, and is consequently merged could be assigned to a number of subspaces. To remove this ambiguity, if one of the detections is frontal its assigned only to the frontal subspace. If none of the detections are frontal it is then assigned to the left-profile subspace.

### 5.4 Face Normalisation

Prior to applying the feature extraction methods outlined above, all images (probe, gallery, and training) are subject to photometric and geometric normalisation.

#### 5.4.1 Eye Detection (Geometric)

Geometric normalisation ensures that the faces we are comparing are of the same scale, rotation and position. Due to the symmetric nature of the face, we use the position of the eyes on the face and apply the appropriate transformations to align the face as required.
The eyes can be annotated manually via the UI, or automatically detected. For automatic detection we use the MPT[21] library which provides training data for eye detection. Once the eyes are marked, the image is subject to a transformation matrix, which scales, translates, and rotates the image to bring the eyes to a fixed position.

Figure 5.4: An outline of the geometric normalisation using annotation of the eyes

5.4.2 Bayesian Face-Sub regions (Photometric)

We use the normalisation algorithm presented by Gross & Brajovic[16], which solves an optimization problem to estimate the illuminance map of the image. The reflectance map is evaluated using this estimate, based on which the image is photometrically normalised.

To provide this functionality, the Torch3 and Torch3Vision[22] libraries were ported from Linux to Windows. These libraries provide the 2 implementations to Gross & Brajovic’s algorithm which vary only but the manner in which they solve the optimisation problem to estimate the illumination map of the image.
Chapter 6

Application To Video

A video can be thought of simply as a sequence of images, with temporal dependencies. The system was initially developed with still probes in mind, and was later extended for video sequences. In this chapter I discuss this extension and what it entailed.

6.1 Overview

In order to extend our system to video, we had to provide the following further functionality:

- Tracking of detected faces (as discussed in Chapter 3)
- A model for applying the identification algorithm to multiple images of a subject
- Additional UI functionality and extended system architecture

The UI functionality is not discussed. Extensions to the architecture are provided in Chapter 6.

6.2 A Model for Identification

6.2.1 Frame selection algorithms

With video probes, we no longer have a single face for a subject but potentially a large number of faces which are obtained over a number of frames. As we know all of the faces belong to the same subject, we need not identify each of the segmented faces individually. The purpose of the frame selection algorithm, is to attempt to identify the subject in question using the least number of faces.
The methods implemented for this purpose, attempt to select a single face from the complete set of faces (for a given subject), which is believed to be the suitable for identification purposes.

**Select First**

This method ignores all consequent tracked faces and performs recognition on the initial detected face. The justification behind this approach is that detector is specifically trained to detect faces, as opposed to the tracker which is trained to follow regions of a certain colour. Therefore it is reasonable to believe that the initial face detected is suitable if not the best face out of all the segmented faces for a given subject. In addition we have pose information about the face.

Although this method may work well, this is more to do with the shortcomings of the tracking component as well as the inability of the detection component to detect faces which are not frontal or profile.

**Minimise Geometric Normalisation**

Ideally the face we are attempting to identify will be aligned with all images in the gallery. If it is not we use the annotation of the eyes to attempt and geometrically normalise to achieve the correct alignment. However when normalising it the image obtained can be degraded which may cause problems for identification. We form a novel approach, where we select the frame which requires the least amount of geometric normalisation.

Normalisation is done by detecting the eyes, from which the image is scaled, translated, and rotated. Each face is assigned a penalty score, based on the amount of scaling, translation and rotation performed.

The three transformations have values associated with them, which are themselves normalised between 0 and 1. The final penalty is a weighted sum of all three normalised transformations, where scaling is penalised the most followed by translation and rotation. We then select the face yielding the minimal penalty score for identification.
Chapter 7

Engineering *enVisage*

This chapter provides an overview of the overall design and architecture of the underlying system, that makes enVisage.

### 7.1 Overall Architecture

The system design grew from what the original diagram giving an overview of the recognition flow. Most importantly the definition of recognition being a combination of segmentation and identification is carried through as shown in the overall system architecture is shown below:

![Overall architecture of enVisage](image)

*Figure 7.1: Overall architecture of enVisage*

I will now proceed, by providing a summary of the responsibilities and internal design of each of these components. An extensive system design outlining every class and method is not provided, and it can be inspected in the source code made available at [http://www.doc.ic.ac.uk/ av102/envisage/](http://www.doc.ic.ac.uk/ av102/envisage/)
7.1.1 System Core

This is program loading class, which creates instances of the 4 managers. All communication between managers is performed via the Core. In addition it is responsible for all I/O operations of the system, including:

- Loading / Saving Gallery
- Loading / Saving Training Data
- Loading of Probes

All probe images are stored in the core, as they interact with all managers but the Gallery, thus minimising the chain of command.

In addition the core holds the functionality to coordinate recognition in video sequences. For this purpose it maintains a list of frame selection algorithms, which are required to fully automate recognition. It should be noted, that all system properties are held in a static Properties class, allowing global access to get and set the properties.

7.1.2 User Interface

A thin client approach is adopted, where all of the functionality of the system is shifted away from the front end. Thus it is the sole responsibility for the UI is to provide a view to the access functionality of the system. All requests are passed to the System Core, which is then delegated to the appropriate manager.

7.1.3 Identification Manager

The Identification Manager provides access to training, feature extraction and matching, as well as preprocessing algorithms.

Training, feature extraction and matching are encapsulated in concrete classes implementing IdentificationAlgorithm. Similarly preprocessing algorithm are those implementing Preprocessing. This manager maintains aggregations of concrete instances of both algorithms. In addition all training data formed by any IdentificationAlgorithm, is stored in the Manager, who maintains Training Data for each algorithm, with each gallery.
Figure 7.3: Two screenshots of enVisage User Interface. Left image is probe loading pane. Right image is the Gallery administration pane.

Figure 7.4: UML Diagram outlining Design of Identification Manager
7.1.4 Segmentation Manager

Given a video or image probe, the segmentation manager is responsible for detecting and tracking (if necessary) all faces in the scene.

It maintains a collection of algorithms for detection and tracking, which it utilises to obtain the locations of faces. In addition the segmentation manager is also responsible for producing the segmented face image from the scene, based on the results of detection and tracking.

7.1.5 Gallery Manager

The Gallery Manager is responsible for maintaining the database of stored faces (i.e. the Gallery). All update procedures to the database are provided via the manager, which performs the necessary actions on the Gallery.
7.2 Technology

The system we see before us today utilises a number of technologies, as shown below.

C# was chosen as the primary technology which was used to realise the above architecture. This was chosen for reasons outlined below:

- Similar to Java offering automatic garbage collection and based on Object Oriented Paradigm, which the developer is experienced in.

- A large number of additional language features, which aid fast and easy development

- Ability to override garbage collection if high performance is required

- Easy to build Graphical User Interface

- The libraries for components not implemented in the project (e.g. Detection, Tracking etc) were predominantly in C++. C# offers seamless interoperability with external C++ libraries.

In addition, XML was used to store ground truth relating to the gallery images, as support to handle and manipulate XML databases are readily in C#, and as they are 'self describing' in their structure.
7.3 Design Methodology

The software was developed and tested using Microsoft Visual Studio 2005 on a Pentium 2.53Ghz, with 512 RAM. Memory performance and potential leaks were tested using .NET Memory Profiler.

A test harness was developed to carry out large scale experimentation and testing to obtain the results presented in the evaluation.

In accordance with the eXtreme Programming philosophy, the software was developed in boxed iterations of length 2 weeks. Iterative development is low risk due to earlier risk discovery and mitigation. Changes required at the core were invoked in early iterations, and a partial product was always available. Time boxing also ensured that we focussed on the end date, rather than letting Parkinson's law kick in. Small iterations also meant that smaller problems of lower complexity were tackled at each stage.
Chapter 8

Testing & Evaluation

This chapter presents the quantitative analysis carried out using the system described in the previous chapters.

As a reminder, a face recognition system is made of two main components: Segmentation and Identification. The primary aim of this project was to investigate and implement a novel identification algorithm, which was invariant to changes in pose and illumination. Hence the main focus of our testing and evaluation lies in the performance of this algorithm. In addition, the segmentation component is also tested independently, enabling us to evaluate the libraries used for these purposes.

A ad hoc testing harness was written, to facilitate automated testing using large datasets. Results were written to a .csv file, which were then manipulated and analysed using a Microsoft Excel.

8.1 Identification Algorithm

We evaluate both benchmarking and identification performance using both base algorithms: Eigenfaces and Bayesian Eigenfaces. Benchmarking tests are carried out initially for the training and testing phases of both algorithms.

Identification performance, for variations in pose and illumination are assessed independently using separate datasets. The novel algorithm implemented has other components on top of the base algorithms. The impact of these components on performance is also tested, where appropriate.
8.1.1 Benchmarking

We compare the performance of the two base identification algorithms, that is Eigenfaces and Bayesian Eigenfaces. Both algorithms consist of a training and testing phase, and the performance of each is considered separately. All tests were performed on a 2.61Ghz Pentium IV, with 1GB RAM.

Training Performance

Training is the phase of the algorithm, which takes a set of images and obtains a richer and more salient representation for faces. We compare the respective performances of each algorithm, with an increasing number of images in the training set.

![Figure 8.1: A comparison of training times for both base algorithms](image)

For Bayesian Eigenfaces, having \( n \) images of the same subject is equivalent to \( ^nC_2 \) training images. Therefore training performance was evaluated for \( n = 2 \ldots 15 \) images, which is equivalent to \( 1 \ldots 105 \) training images. Linear regression was used to interpolate between the values.

As we see above in Figure 8.1, training using Bayesian Eigenfaces is less time consuming than Eigenfaces. This may be attributed to fact, that to perform training on \( n \) images using Bayesian Eigenfaces, requires much less images to be loaded from disk (as
shown above). In comparison Eigenfaces will have to load all \( n \) images, which is time consuming.

**Testing Performance**

In the testing phase, both base algorithms project the probe image onto the face space extracted during training. The probe’s feature vector is then compared with all stored feature vectors of gallery images. We compare the performance of each algorithm, with an increasing number of images in the gallery.

![Figure 8.2: A comparison of the Testing phase for both base algorithms](image)

As seen in Figure 8.14, the Bayesian algorithm is slightly faster (\( \simeq 1.31 \text{ ms} \) on average) than the PCA algorithm. The only difference computationally between these algorithms, is that PCA has an additional step of subtracting the mean image from the probe before projection, which this small performance gain could be attributed to.

### 8.1.2 Identification Performance

The identification algorithm implemented is made of many components, each of which were justified (in theory) in Chapter 3 of the report. These include:

- Illumination Normalisation using Gross & Brajovic’s Algorithm
• Geometric Normalisation using annotation of the eyes
• Basic Eigenfaces algorithm
• Bayesian Eigenfaces
• View Based Eigenfaces

The aim of this testing is to verify that each of the above parts do in fact impact the identification performance, ideally in a positive manner. From our results we should be able to ascertain which of the techniques explored should be used to form the best identification algorithm, that is invariant to pose and illumination, and comment on its performance.

Notation
A gallery \( G \) consists of biometric samples \( \{g_1, \ldots, g_{|G|}\} \).

A probe set \( P \) consists of biometric test samples \( \{p_1, \ldots, p_{|P|}\} \), where \( p_j \in P_G \) if the probe is represented in the gallery, otherwise \( p_j \in P_N \).

\( s_{ij} \) - Similarity score between \( g_i \) and \( p_j \) (where \( g_i \in G \) (\( G \) is gallery set) and \( p_j \in P \) (\( P \) is probe set)) \(^1\)

If \( p_j \) is a biometric sample of a person in the gallery, let \( g^* \) be its unique match in the gallery. The similarity score between \( p_j \) and \( g^* \) is denoted as \( s_{*j} \)

A probe \( p_j \) has rank \( n \) if \( s_{*j} \) is the \( n^{th} \) largest similarity score, between \( p_j \) and all gallery images, denoted \( \text{rank}(p_j) = n \). Rank 1 is sometimes called the top match.

Closed-set Identification
Performance is tested and compared using Closed-set Identification.
In a closed set, all probe subjects are represented in the gallery. As we know at least one correct match exists for each probe, the question we are asking here is "Is the correct answer in the top \( n \) matches?" (where the top match is deemed as the nearest face).

In the case where the probe is represented many times in the gallery, a highest ranking correct answer is taken. Ideally the the top match would always provide the correct identity of the probe.

\(^1\)Larger scores indicate that the two biometric samples are more similar (2) the score is a match score if the identity of the samples are the same person; if not it is referred to as a non-match score.
Once testing is performed as described above, the performance is reported on a cumulative match characteristic (CMC). A CMC plots the identification rate ($P_I(n)$) as a function of rank $n$. The identification rate for rank $n$, $P_I(n)$, is the fraction of probes at rank $n$ or lower. For rank $n$, let

$$C(n) = |\{p_j : \text{rank}(p_j) \leq n\}|$$

be the cumulative count of the number of probes of rank $n$ or less. The identification rate at rank $n$ is

$$P_I(n) = \frac{|\{p_j : \text{rank}(p_j) \leq n\}|}{P_G}$$

The functions $C(n)$ and $P_I(N)$ are nondecreasing in $n$. The identification rate at rank 1, $P_I(1)$, is also called the correct identification rate and the top match rate or score.

**Details Of A Test**

Each test carried out consists of the following details:

- Database of Images
- Images used for Training
- Which images make up the Gallery
- The images used as Probes
- Details of algorithm used

**Invariance against Pose**

There are three components which are included in our algorithm design, which will impact performance with variations in pose, which are:

1. Eigenfaces OR Bayesian Eigenfaces
2. View Based (VB) OR Non View Based (NVB)
3. Geometrically normalised (GN) OR No Geometric Normalisation (GN)

We test each combination of the above three parameters, giving us eight tests to perform. Each test is carried out with the following data:
• **UMIST Face Database**[23]
  Consists of 564 images of 20 people. Each covering a range of poses from profile(+90°) to frontal(0°) views. No ground truth is available for exact pose of subject, therefore it was manually annotated. Ground truth relating to the image's subject id is available via the filename.

• **Training Set**
  Training is performed using 10 subjects, with 4 images per subject at the following approximate poses: 0°, +30°, +60°, +90°.

• **Gallery**
  All 20 subjects are represented in the gallery with 2 images at 0°, and +90° pose.

• **Testing Set**
  All images in the database are used for testing, except those present in the Gallery.

![Training Set](image1)

![Gallery Images](image2)

**Figure 8.3:** Images used for each subject for the Training phase and the Gallery, for Pose Invariant Testing

![Variety of probes](image3)

**Figure 8.4:** Variety of probes used for pose invariant testing.

The line graphs in Figures 8.5 & 8.6, show how using a view based architecture and geometric normalisation impact the performance on both base algorithms.
Figure 8.5: Comparing the use of a View Based Eigenspaces and Geometric Normalisation with EigenFaces(PCA) using the UMIST\cite{23} Face Database.

Figure 8.6: Comparing the use of a View Based Eigenspaces and Geometric Normalisation with Bayesian EigenFaces using the UMIST\cite{23} Face Database.
Table 8.1: Comparing PCA & Bayesian Eigenfaces an Identification Rate at Rank 1 (VB = View-Based, NVB = Non-View-Based, GN = Geometrically Normalised, NGN = Not Geometrically Normalised)

<table>
<thead>
<tr>
<th>Method</th>
<th>VB&amp;GN</th>
<th>NVB &amp; GN</th>
<th>VB &amp; NGN</th>
<th>NVB &amp; NGN</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCA</td>
<td>0.439</td>
<td>0.264</td>
<td>0.4945</td>
<td>0.307</td>
</tr>
<tr>
<td>Bayesian Eigenfaces</td>
<td>0.341</td>
<td>0.274</td>
<td>0.604</td>
<td>0.516</td>
</tr>
</tbody>
</table>

We see a consistent improvement in the performance with both base algorithms, when the view based architecture is used and the images are *not* subject to geometric normalisation based on the location of the eyes.

An improvement in performance using the view based architecture is as expected, however the deterioration in performance due to geometric normalisation is unexpected.

![Original Image](image1) ![Normalised Image](image2)

**Figure 8.7:** Problems with geometric normalisation: Notice the unset pixels in the bottom left corner of (b)

On closer inspection, I believe that this discrepancy may be to do with the manner in which images are geometrically normalised. It is noticeable from Figure 8.7 that normalised images sometimes result with part of the image not being defined (i.e. pixels not set), which results in being set to black.

As a result, these areas in comparison to their respective areas in other images, show large variation which may be extracted in the feature extraction process. However they are of no use for identification, and may as seen in our results, only hinder it.

The transformation applied to achieve geometric normalisation, is simply rotation, scale, and translate. A more complex transform may be used which may possibly skew the image, to avoid pixels not being set. Alternatively, pixels which are not set could be set to a local average of nearby pixels or a global pixel average.
Summary:

- Bayesian Eigenfaces shows an increase in performance over PCA in all cases, bar one (i.e. View based & Geometrically Normalised). On average the performance gain was a 5% increase in Rank 1 identification.

- A View Based Architecture also increases performance in all cases. On average the performance gain was a 12% increase in Rank 1 identification.

- Geometric Normalisation resulted in a decrease in performance in all cases. On average the performance drop was a 15% decrease in Rank 1 identification.

Invariance against Illumination

Performance changes due to variation in illumination, will be impacted by the following two components of our algorithm:

1. Eigenfaces OR Bayesian Eigenfaces
2. Using Gross & Brajovic Normalisation OR No Normalisation

We test each combination of the above two parameters, giving us four tests to perform. Each test is carried out with the following data:

- **YALE Face Database [24]**.
  Consists of 2443 images of 39 subjects. Each subject is captured at frontal pose, under 64 variations in illumination. All subjects are geometrically normalised. Ground truth relating to the illumination is specified as light source direction (e.g. A+000) with respect to the camera axis as well as elevation (e.g. E+040). Corrupt images, or those which were totally black were removed from the dataset, prior to testing.

- **Training Set**
  Training was performed using 15 subjects, with 2 images per subject (i.e. A+000E+00, A+020E-40).

- **Gallery**
  All subjects are represented in the Gallery with a single image (i.e. A+000E+00).

- **Testing Set**
  All images in the database are used for testing, except those present in the Gallery.
Figure 8.8: Images used for each subject for the Training phase and the Gallery, for Illumination Invariant Testing.

Figure 8.9: Variety of probes used for Illumination Invariant testing.

Figure 8.10: Comparing the performances of the base algorithms with and without photometric normalisation on the YALE face database.
### Table 8.2: Comparing PCA & Bayesian Eigenfaces and Identification Rate at Rank 1 (VB = View-Based, NVB = Non-View-Based, GN = Geometrically Normalised, NGN = Not Geometrically Normalised)

<table>
<thead>
<tr>
<th></th>
<th>No Normalisation</th>
<th>Gross &amp; Brajovic Normalised</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCA</td>
<td>0.3979</td>
<td>0.6281</td>
</tr>
<tr>
<td>Bayesian PCA</td>
<td>0.4099</td>
<td>0.7310</td>
</tr>
</tbody>
</table>

**Summary:**

- Bayesian Eigenfaces performs better than Eigenfaces in both cases, showing a 5% increase in performance, on average.

- Preprocessing images with Gross & Brajovic’s photometric normalisation algorithm, increased performance by approximately 27% on average, for both base algorithms.

#### 8.1.3 An Ideal Algorithm

The best performing algorithm from the testing carried out above is:

- **Pose**: Bayesian Eigenfaces, using a View Based Architecture.

- **Illumination**: Bayesian Eigenfaces using Gross & Brajovic preprocessing algorithm.

Both of the above returned identification rates of 60% and 73% respectively. Therefore from our testing we can conclude, to achieve the best recognition rates across pose and illumination, our novel algorithm should be as shown in Figure 8.11 (Note: Geometric Normalisation has been removed).

#### 8.2 Face Detection

To segment faces from any probe, we are reliant on Intel CV’s Haar based Object Detection Algorithm. Using this library, a multi-view face detector has been implemented, which is able to detect faces with out of plane rotation ranging from $-90^\circ$ to $+90^\circ$.

#### 8.2.1 Detection Performance

We test the performance, using the CMU Frontal & Non-frontal face databases[25]. Images showing some of the results of the testing are shown in Figures 8.12 & 8.13.
Figure 8.11: A novel algorithm, formed by combining the best performing components for variation across Pose and Illumination.

Figure 8.12: Face detection results from the CMU Non-Frontal Test set
Figure 8.13: Face detection results from the CMU Frontal Test set (images from all subsets included)

Figure 8.14: Detection Rates for each of the CMU Datasets
<table>
<thead>
<tr>
<th></th>
<th>True Positives</th>
<th>True Negatives</th>
<th>False Positives</th>
<th>Total Faces</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-Frontal</td>
<td>222</td>
<td>227</td>
<td>47</td>
<td>449</td>
</tr>
<tr>
<td>Frontal (New Test)</td>
<td>161</td>
<td>24</td>
<td>29</td>
<td>185</td>
</tr>
<tr>
<td>Frontal (Rotated)</td>
<td>52</td>
<td>171</td>
<td>30</td>
<td>223</td>
</tr>
<tr>
<td>Frontal (Test)</td>
<td>127</td>
<td>42</td>
<td>10</td>
<td>169</td>
</tr>
<tr>
<td>Frontal (Test Low)</td>
<td>114</td>
<td>43</td>
<td>7</td>
<td>157</td>
</tr>
</tbody>
</table>

Table 8.3: Detection Results on CMU Face Databases (Frontal database has 4 subsets - "Rotated" are frontal faces with out of plane rotation, "Test Low" are low resolution images, "Test" & "New Test" both contain a variety of frontal images with no other variation

**Summary:**

The face detector produced, performs well across all frontal faces (average detection rate of 78%) except those exhibiting rotation (23%). A detection rate of 49% is exhibited across images containing profile faces.
Chapter 9

Conclusion

This project started with the following objectives:

- Implement a novel algorithm for identifying an individual, subject to changes in pose and illumination
- Implement an algorithm to segment faces from video/still images, using open source libraries
- Engineer software to unify all of the above, into a recognition system

9.1 What has been achieved?

9.1.1 A Novel Algorithm

It was our aim to implement a novel algorithm, which integrated a number of techniques specified in current research, to achieve invariance to both changes in pose and illumination.

I approached this objective by carrying out extensive research in the area. The knowledge instilled as a result of this was then documented in a literature review, provided in Chapter 2.

Using this knowledge, two subspace based feature extraction techniques have been implemented (Eigenfaces and Bayesian Eigenfaces). An architecture supporting View-based Eigenfaces, was also developed, and annotation of the pose of a probe was acquired at the detection stage.

Photometric normalisation, using Gross & Brajovic’s algorithm was integrated, using a ported library. A library performing automatic annotation of the eyes was also utilised, and geometric normalisation was performed on
the basis of it.

A number of different algorithms integrating combinations of the above technologies were tested. Performance against changes in pose and illumination were evaluated independently using large scale face databases.

The optimal combination of the techniques specified above, provided an identification rate of 60% and 73% for changes in pose and illumination respectively.

From the work carried out, there is one important lesson I have learnt about this identification problem:

*Variation can be both your best friend as well as your worst enemy.*

We need variation, as that is what enables us to distinguish one person from another. However *variable variation* (e.g. images of different subjects which vary in their pose), usually results in using the wrong information to distinguish between individuals, and performance deteriorates drastically.

As with most problems faced in life, the best way to tackle it, is to break it down. This idea is essentially what a view based architecture does, where the problem is broken into two subtasks: identifying frontal faces, and identifying profile faces.

Similarly with Bayesian Eigenfaces, we don’t rely on a single subspace to describe all faces. We essentially split the problem in half again, and form a subspace for changes between the same subject and those between different subjects.

The key point to note is, although faces do inherently exhibit this *variable* variation, we can reduce the *variability* in variation by breaking the problem down. Further work related to this is notion is suggested in the next section.

### 9.1.2 Segmentation Faces

Segmentation is a prerequisite for Identification. If we can’t find faces, we definitely cannot attempt to identify them.

Although this was not the focal point of the project, a multi-view face detector and colour based tracking have been integrated to segment faces from still and video probes.

Although the algorithms were not implemented from scratch, the theory behind them was studied, understood, and summarised in Chapter 3.
In addition the the performance of the detection component was evaluated using the CMU frontal and non-frontal databases. Frontal faces which are not rotated yielded a detection rate of 78%, whilst profile faces were detected 49% of the time.

9.1.3 enVisage

The vision of a overall recognition system unifying the above novel identification and segmentation components was realised with enVisage.

The design was closely aligned to the flow of recognition, such that anyone who is familiar with recognition should easily be able to navigate around the code, and add their own implementations of detection, tracking, identification etc.

The UI, provides, manual means for segmentation of faces, annotation of eyes, and selection of frame(s) for identification for video probes. Therefore, it provides the ideal platform for anyone looking to develop a certain component of the recognition flow, whilst not worrying about the other interdependent components.

All functionality of administration of the Gallery, is provided and multiple Galleries are also supported, which was used for View Based Eigenspaces.

9.2 Further Work

Additional work is suggested in two areas (1) Improvements to enVisage, the software (2) Further Investigation for Identification across Pose & Illumination.

9.2.1 Improvements to enVisage

The following are improvements, that can be made to the underlying software whose role it is to unify the various components of recognition:

- Program to Product
  Usability, Branding, User Support etc. were not the focus of this project. It was not our objective to produce shrink wrapped recognition software. However, what we do have is a robust program, with a neat, simple interface. I believe with more time being spent on the GUI, and with the appropriate branding (i.e. logo, theme etc), enVisage could be transformed from the program it is today to a marketable product of tomorrow.
• **Realtime Recognition**
Currently recognition for video probes can only be done statically (i.e. for a fixed length file from disk). All processing is done at once, before presented to the user. An extension to this would be to provide realtime recognition through a capture device such as webcam. A concurrent model for processing would be required, where frames are processed as they appear and presented to the user in realtime.

• **enVisage Live!**
Much of my research for this project started on the web, from which I came across a couple of exciting websites: riya.com and betaface.com. Riya is a startup company, which is the first to use face and text recognition with online photo albums. Betaface also provides face recognition for still images via the web, and even offers access to its services via a webservice.

With the recent surge of video media on the web (e.g. video.google.com etc.), providing similar recognition services such as riya and betaface for video probes would provide a challenging and extremely useful extension.

### 9.2.2 Enhancing Identification across Pose & Illumination

The following suggestions, are areas of investigation based on what has been learnt from this project, to improve recognition invariant to changes in pose and illumination.

• **Investigating Sub Problems**
In the previous section, I talked about the problem of variable variation, and how it was reduced by breaking the problem down, using techniques such as View-Based Eigenspaces. Currently, this idea has been used but in a limited capacity, which has been restricted to breaking the problem down by Pose, intrapersonal and extrapersonal variations, or specific features (eyes, mouth etc).

This same idea can be used for a variety of attributes, and we could break the identification problem by properties such as sex, age, ethnicity, and potentially combine them. If we form a more specific subspace (e.g. a subspace particularly for asian women under the age of 25), the intrinsic manifold would be better suited for the specific category of faces it represents, which would improve identification.

• **Non Linear Bayesian Eigenfaces**
As mentioned in the Chapter 2, due to the complex nature of faces the intrinsic manifold is highly unlikely to be linear. PCA is a limiting approach as it extracts a linear manifold with orthogonality constraints. Therefore, the manifold extracted by its application would not extract
the most salient representation of the face. In Bayesian Eigenfaces, we are finding the manifold for differences between faces, which again is nonlinear, hence using PCA to form the intrapersonal subspace as I did, could be improved upon by using a Non-Linear Subspace technique such as NLPCA.
Bibliography


